⑲日本国特許庁(JP)　⑪特許出願公開

⑫公開特許公報(A)　昭63-192135

⑤Int.Cl.⁴	識別記号	庁内整理番号	㊸公開　昭和63年(1988)8月9日
G 06 F 9/30	350	G-7361-5B	
9/44	310	8724-5B	

審査請求　未請求　発明の数 6　(全30頁)

㊹発明の名称　信号処理方法、信号プロセッサ、その設計方法及びマイクロプロセッサ

㉑特　願　昭62-317311
㉒出　願　昭62(1987)12月15日
優先権主張　㉜1986年12月15日㉝米国(US)㉛941,450

㉗発明者　グレゴリー・アラン・ポータノヴア　アメリカ合衆国コネチカット州、エンフイールド、アーバー・ロード　50
㉗発明者　ブライアン・ジヨセフ・スプラギユー　アメリカ合衆国コネチカット州、エンフイールド、バーチウツド・ロード　18
㉑出願人　ユナイテツド・テクノロジーズ・コーポレイション　アメリカ合衆国コネチカット州、ハートフオード、フイナンシヤル・プラザ　1
㉔代理人　弁理士　明石　昌毅

明　細　書

1．発明の名称

信号処理方法、信号プロセッサ、その設計方法及びマイクロプロセッサ

2．特許請求の範囲

(1) 信号処理方法に於いて、

別々のバスを経て縮小命令集合命令及びデータを処理する過程と、

前記データバスを経て複合命令及びデータを処理することにより、またエミュレートされた各複合命令に対する縮小命令の複数個の群であって各群が縮小命令の特定の複合命令に対応する縮小命令の複数個の群の一つを処理することにより、複合命令集合信号処理方法をエミュレートする過程と

を含んでいることを特徴とする信号処理方法。

(2) 複合命令集合命令に応答する信号プロセッサの設計方法に於いて、

縮小命令集合信号プロセッサとして使用するため別々のデータ及び命令バスを有する縮小命令集合信号プロセッサを設計する過程と、

前記縮小命令集合信号プロセッサのなかへエミュレーション能力を設計する過程とを含んでおり、それにより前記データバスが到来する複合命令集合命令及びそれに関係する双方向データの双方に対して使用され、また前記命令バスが縮小命令集合命令のみを受信するために使用され、複数個の縮小命令集合命令が受信された各到来複合命令に対して実行されることを特徴とする信号プロセッサの設計方法。

(3) 信号プロセッサに於いて、

縮小命令集合信号プロセッサとして使用するため別々のデータ及び命令バスを有する縮小命令集合信号プロセッサと、

前記信号プロセッサを複合命令集合命令信号と前記データバスを経て受信されたデータ信号とに応答するようにするため、また前記データバスを経てデータ信号を供給するためのエミュレータ手段とを含んでおり、前記エミュレータ手段が前記信号プロセッサを、受信された各複

合命令集合命令信号に応答して、前記縮小命令集合信号プロセッサ命令アドレスバス上に縮小命令アドレスを供給するようにさせ、また、供給されたこのような各縮小命令アドレスに対して、前記信号プロセッサをそのエミュレーション中に実行するための一連の対応する縮小命令を供給するようにさせることを特徴とする信号プロセッサ。

（4）別々のデータ及び命令バスを有する信号プロセッサに於いて、

第一の入力信号及び第二の入力信号に論理演算を実行するため、また前記論理演算の結果を示すＡＬＵ出力信号を供給するため第一の入力信号及び第二の入力信号に応答する算術論理演算装置（ＡＬＵ）と、

前記ＡＬＵ出力信号を記憶及び供給するため前記ＡＬＵ出力信号に応答するアキュムレータと、

前記ＡＬＵ出力信号又はデータバスからの到来オペランド信号を選択された記憶レジスタのなかに記憶するため、宛先入力信号を供給するため、また源入力信号を供給するため、前記ＡＬＵ出力信号又は前記オペランド信号に応答する、複数個の記憶レジスタを有するレジスタファイルと、

前記ＡＬＵに論理演算の実行のために供給するため、もしくはデータバス上に出力オペランド信号として供給するため、選択された記憶レジスタの信号内容を宛先レジスタ出力信号として供給する以前に前記信号内容を記憶するため、前記宛先入力信号に応答する宛先レジスタと、

前記ＡＬＵに論理演算の実行のために供給するため、もしくはデータバス上に出力オペランド信号として供給するため、選択された記憶レジスタの信号内容を記憶レジスタ出力信号として供給する以前に前記信号内容を記憶するため、前記源入力信号に応答する源レジスタと、

次の命令をアドレス指定するためそのアドレスをインクレメントするためインクレメントする信号に応答し、又は各々命令アドレスバス上に供給される例外命令アドレス信号を記憶及び供給するため前記ＡＬＵ信号に応答する命令アドレスカウンタと、

前記命令信号を記憶及び供給するため命令バスからの命令信号に応答する命令レジスタと、

信号プロセッサを制御するため命令信号を復号するため前記命令レジスタからの前記命令信号に応答する制御手段と、

前記源レジスタ出力信号及び前記命令信号に、また前記第一のＡＬＵ入力信号を供給するため選択信号に応答する第一のマルチプレクサ手段と、

前記宛先レジスタ出力信号、前記アドレス信号及び前記アキュムレータからの前記ＡＬＵ出力信号に、また前記第二のＡＬＵ入力信号を供給するため選択信号に応答する第二のマルチプレクサ手段と

を含んでいることを特徴とする信号プロセッサ。

（5）別々のデータ及び命令バスと共に使用するため機械サイクルあたり少なくとも二つのクロック周期を有する本質的に一つの機械サイクルの命令集合信号プロセッサに於いて、

次の命令をアドレス指定するため内部に記憶されている現在のアドレス信号をインクレメントするため各機械サイクルの第一のクロック周期の第一の部分の間に供給されるインクレメントする信号に応答し、また例外命令をアドレス指定するため命令アドレスバス上の例外命令アドレス信号を供給するため内部に記憶するため各機械サイクルの第一のクロック周期の第一の部分の間にＡＬＵ出力信号に応答する命令カウンタと、

各機械サイクルの第二のクロック周期の第二の部分で開始し且つそれに続く前記命令信号を受信、記憶及び供給するため、命令カウンタにより第一のクロック周期の第一の部分の間にアドレス指定された命令信号を置換し且つ第二のクロック周期の第二の部分の開始以前に命令バス上に送るべく、第二のクロック周期の第二の

部分の間に応答する命令レジスタと、

各機械サイクルの第一のクロック周期の間に前記命令レジスタ信号を復号するため、また各機械サイクルの第二のクロック周期の間に信号プロセッサに対する制御信号を供給するため、先の機械サイクルの第二のクロック周期の第二の部分の間に前記命令レジスタのなかに記憶された前記命令信号に各機械サイクルの第一のクロック周期の間に応答する制御手段と、

論理演算を実行するため、また次の機械サイクルの第一のクロック周期の第一の部分の間の前記論理演算の結果を示すＡＬＵ出力信号を供給するため、第一の入力信号と、第二のクロック周期の第一の部分で開始しまた続いて供給される第二の入力信号とに選択された機械サイクルの間に応答する算術論理演算装置（ＡＬＵ）と、

前記ＡＬＵ出力信号を記憶及び供給するため、選択された機械サイクルの第一のクロック周期の第一の部分の間に前記ＡＬＵ出力信号に応答するアキュムレータと、

選択された記憶レジスタのなかに前記ＡＬＵ出力信号及び前記オペランド信号を記憶するため、選択された機械サイクルの第一のクロック周期の第一の部分の間に前記ＡＬＵ出力信号に応答し、もしくは選択された機械サイクルの第一のクロック周期の第一の部分の間にデータバスからの到来オペランド信号に応答し、また宛先入力信号を供給するためまた記憶レジスタを選択するため第一のクロック周期の第二の部分の間に記憶レジスタを選択するため、また源入力信号を供給するため第一のクロック周期の第二の部分の間に記憶レジスタを選択するため、前記制御手段からの前記制御信号の一つ又はそれ以上に応答する、複数個の記憶レジスタを有するレジスタファイルと、

選択された記憶レジスタのなかに信号情報を記憶するため、また前記ＡＬＵにより演算されるべき宛先出力信号としてもしくはデータバス上の出力オペランド信号として使用するため第二のクロック周期の第一の部分で開始する宛先出力信号として前記記憶された情報を供給するため、前記宛先入力信号に応答する宛先レジスタと、

選択された記憶レジスタのなかに信号情報を記憶するため、またいずれも第二のクロック周期の第一の部分で開始して供給される、前記ＡＬＵにより演算されるべき源出力信号もしくはオペランドアドレスバス上の出力オペランドアドレス信号として前記の記憶された源信号を供給するため、前記源入力信号に応答する源レジスタと、

第一の選択信号に応答して、第二のクロック周期の第一の部分の間に始動する前記ＡＬＵに前記第一の入力信号として前記源出力信号もしくは前記命令信号を供給するため、選択された機械サイクルの間に、前記源レジスタにより供給される前記源出力信号と先の機械サイクルの間に記憶された前記命令信号とに応答する第一のマルチプレクサ手段と、

第二の選択信号に応答して、第二のクロック周期の第一の部分の間に始動する前記ＡＬＵに前記第二の入力信号として前記アキュムレータからの前記ＡＬＵ出力信号、前記宛先出力信号もしくは前記命令アドレス信号を供給するため、選択された機械サイクルの間に、前記アキュムレータからの前記ＡＬＵ出力信号と前記宛先により供給される宛先出力信号と前記命令カウンタのなかに記憶された前記命令アドレス信号とに応答する第二のマルチプレクサ手段と

を含んでいることを特徴とする信号プロセッサ。

（６）縮小命令集合マイクロプロセッサに於いて、

外部命令バスから命令信号を受信するための命令バス入力ポートと、

データ信号をそれぞれ外部データバスから受信し且つそれへ供給するためのデータポートと、

データ信号をそれぞれデータポートから探索し且つそれへ供給するため前記命令信号に応答する中央プロセッサユニットと

を含んでいることを特徴とするマイクロプロセッサ。

3．発明の詳細な説明

技術分野

本発明は計算機、一層詳細には縮小命令集合計算機（Reduced Instruction Set Computer）（ＲＩＳＣ）に係る。

背景技術

複合命令集合計算機（Complex Instruction Set Computer）（ＣＩＳＣ）の提案者は以前にはソフトウェアのみによりなされた機能をするのに高度に複雑なマイクロプロセッサハードウェアを構成するべくますます超大規模集積回路を使用する。これはソフトウェアがますます高価になり、またハードウェア（ＶＬＳＩ）がますます安価になってきた結果である。より安価でより複雑なハードウェアを使用することにより、ＣＩＳＣ設計者は、より高いレベルの言語がより簡単に、従ってまたより使用し易くなり、またソフトウェア開発費用が低下すると推論する。

他方に於いて、ＲＩＳＣ提案者はより多くの機能がソフトウェアによりなされるより簡単なマイクロプロセッサを創造する。このような機械は所与のプログラムのなかで実行される機能の大多数は、全て一つの機械サイクル内での実行のために設計され得るロード、記憶、比較、分岐、整数算術、論理シフトなどのようなむしろ簡単な機能である傾向を有するという洞察に基づいている。実行されるべき各複合機能に対する特定のＣＩＳＣのアーキテクチュアを解釈するためのマイクロ命令を有する代わりに、ＲＩＳＣ内のアーキテクチュア的アプローチはハードウェアによる直接実行のために設計された縮小命令集合を有していなければならない。従って、マイクロプロセッサアーキテクチュアは、この場合にまさにコードであるマイクロアーキテクチュアに正確にマッチしているので、すなわち制御線をセットするためマイクロコード内に１及び０をセットすることを含む煩雑なプログラミングが存在しないので、必要とされる解釈は存在しない。全てのことがコード内で正確に実行される機能により自動的に行われる。

ＲＩＳＣ設計アプローチには多くの局面が存在し得る。種々の源から抽出されたこのような局面を説明する見事な試みがＩＥＥＥ計算機編、１９８５年９月、第８～１９頁のロバート・ピー・コウウェル他の論文"計算機、複雑さ及び論点"になされた。そこに説明されている局面は、（１）計算機の命令の流れを支配する簡単な機能の迅速な実行を容易にし、また低い解釈的オーバーヘッドを助長するための"単一サイクル演算"、（２）単一サイクル演算の要望から続く"ロード／記憶設計"、（３）マイクロコードはより遅い制御経路に通じまた解釈的オーバーヘッドを増すので、最も迅速に可能な単一サイクル演算を行うための"ハード配線による制御"、（４）制御手段による迅速で簡単な解釈を容易にする"相対的に少数の命令及びアドレス指定モード"、（５）一貫した使用により、再び制御経路を速くするハード配線による復号を容易にするための"固定された命令フォーマット"、及び（６）コンパイラのなかへ静的実行時間を明示的に移動させる機会を提供する"より多くのコンパイル・時間努力"。

上記の文献中に指摘されているように、ＲＩＳＣ及びＣＩＳＣに関する通常の議論は、おそらくそれらの頭辞語のために、議論の領域が機械の命令集合の候補を選択することに制限されるべきであることである。命令の数はＲＩＳＣ文献中の第一の論点の一つであるけれども、ＲＩＳＣ理論の最良の一般化はこの論点を越えている。コルウェル他は、ＲＩＳＣ理論が、いくつかの特殊な文脈のなかで測って、性能を最大化するためアーキテクチュア／実現、ハードウェア／ソフトウェア及びコンパイル時間／実行時間の境界を横切って自由に且つ意識的に設計のトレードオフをする意志を含蓄することを指摘している。

この考えによれば、ＲＩＳＣ及びＣＩＳＣの頭辞語は任意の機械が一方又は他方として分類され得ることを意味するように見えるけれども、実際にはＲＩＳＣ及びＣＩＳＣは連続的な多次元設計空間の異なるコーナーにあるものとして考えられ

るべきである。従って、他方と相容れない一方が必要とされるのではなく、新しいシステムを想像するのに設計者により使用され得るようにＣＩＳＣの経験及びＲＩＳＣの信奉から引き出された技術の組合わせを定型化することが必要とされる。

上記のことにもかかわらず、命令の数がアーキテクチュアをＲＩＳＣ又はＣＩＳＣとして分類するための重要な規範であることは理解されよう。従って、かかる技術を定型化する以下のＲＩＳＣの開示及び特許請求の範囲は、コルウェル他により説明されているようなＲＩＳＣ機械の属性のいくつか又は全てを有するＲＩＳＣに制限することなしに、実際に縮小命令集合を有する計算機に対する設計フィロソフィに属するものとして理解されるべきである。換言すれば、以下の開示はいくつかのＲＩＳＣ構想を引いてはいるが、"設計空間"の"コーナー"にはない。

軍は航空機計算機に対して標準１６ビット複合命令集合アーキテクチュア（ＭＩＬ－ＳＴＤ－１７５０及びその後継）を制定してきた。この標準の目的は、計算機の特殊な実現の詳細を定めることなしに空軍の航空兵器システムの仕様を定めるための均一な命令集合アーキテクチュアを確立することである。従って、それは複合命令集合アーキテクチュアを規定するのみであり、システム独特の要求は各特殊の計算機に対する後日の規定のために残されている。その応用は特定の航空用機能又は特殊のハードウェアに制限されていない。一般に、この標準は適度な精度の航行、計算されたエアーレリーズ点、武器供給、エアーランデブー、貯蔵管理、航空機ガイダンス及び航空機管理のような機能を実行する計算機に、制限なしに、応用可能である。記述"ＭＩＬ－ＳＴＤ－１７５０"はこの文書を通じて、特殊な改訂版が記述されており従ってその意味で理解されるべき場合を除いて、最初の標準及びその後継の全てを記述するのに使用され得る。

ＭＩＬ－ＳＴＤ－１７５０標準の命令集合アーキテクチュアの期待される利点はコンパイラ及び命令レベルシミュレータのような利用可能な支援ソフトウェアの使用及び再使用である。他の利点として、（ａ）武器システム内の二つ又はそれ以上の計算機の使用により得られる全支援ソフトウェアの減少、及び（ｂ）ハードウェアの開発と無関係なソフトウェアの開発のような利点も達成される。

ＭＩＬはプログラマの観点からの機能的操作を規定する。それはデータ形式、命令形式、簡略命令、命令演算、アドレス指定モード、プログラマアクセスレジスタ、割込み構造などを規定する。それは特別な実現を規定しないので、それはベンダー及びテクノロジーに無関係である。上記のように、その使用は標準ソフトウェア支援ツールの使用、マルチベンダー軍用システム中の全支援ソフトウェアの減少、及びハードウェアの開発と無関係なソフトウェアの開発を助長するものと期待される。

ＭＩＬ－ＳＴＤ－１７５０レジスタ集合は１６の１６ビット汎用レジスタ（Ｒ０、…ＲＦ）、１６ビット状態語（ＳＷ）、１６ビット命令カウンタ（ＩＣ）、１６ビットマスクレジスタ（ＭＫ）、１６ビット割込みレジスタ（ＰＩ）及び１６ビット故障レジスタ（ＦＴ）を含んでいる。

支援されるデータフォーマットはバイト（上側、下側）、１６ビット固定点単精度（１６ビットの２の補数）、３２ビット固定点倍精度（３２ビットの２の補数）、３２ビット浮動点（２４ビットの２の補数仮数、８ビットの２の補数指数）及び４８浮動点拡張精度（４８ビットの２の補数、８ビットの２の補数指数）を含んでいる。

ＭＩＬ－ＳＴＤ－１７５０命令集合はこれまでにＣＩＳＣを使用していくつかの会社により実現されてきた複合命令集合である。例えば、なかんずくフェアチャイルド、マクドネル・ダグラス及びパーフォーマンス・セミコンダクターは全て市販されているＭＩＬ－ＳＴＤ－１７５０ＣＩＳＣ機械を有する。

発明の開示

本発明の目的は、簡単なマイクロプロセッサを提供することである。提供されるマイクロプロセ

ッサは縮小命令集合計算機（ＲＩＳＣ）として、又は簡単に簡単なアーキテクチュアを有する信号プロセッサとして見られ得る。"ＲＩＳＣ"という用語は本発明のこの目的と結び付いて本明細書を通じて頻繁に使用される。

本発明の他の目的は、たいていの命令に対して単一サイクル演算を有する縮小命令集合計算機（ＲＩＳＣ）を提供することである。

本発明の他の目的は、単一サイクル演算の上記の目的に付随するＲＩＳＣロード／記憶設計を提供することである。

本発明の他の目的は、迅速な単一サイクル演算のためのハード配線による制御を提供することである。

本発明の他の目的は、制御手段による迅速で簡単な解釈を容易にする比較的少数の命令及びアドレス指定モードを提供することである。

本発明の他の目的は、制御経路を速くするため命令のハード配線による復号を容易にするのに一貫して使用され得る固定されたフォーマットを提供することである。

本発明の他の目的は、多数の可能な縮小命令集合で使用可能な簡単なＲＩＳＣアーキテクチュアを提供することである。

本発明の他の目的は、このような簡単なＲＩＳＣアーキテクチュアの実現に通ずる縮小命令集合を提供することである。

本発明の他の目的は、複合命令集合をエミュレートするため縮小命令集合を使用する信号処理方法を提供することである。

本発明の他の目的は、ＭＩＬ－ＳＴＤ－１７５０をエミュレートし得る縮小命令集合信号プロセッサを提供することである。

本発明の他の目的は、ユーザーがオフチップの記憶及び呼出しのためのＲＩＳＣコードサブルーチンを書くことにより機能を定義し得るＭＩＬ－ＳＴＤ－１７５０マイクロプロセッサを提供することである。

本発明の他の目的は、複合命令集合を効率的にエミュレートするのに使用するための縮小命令集合信号プロセッサを提供することである。

本発明の他の目的は、複合命令に応答するが、それらを縮小命令の群を使用して実行する信号プロセッサの設計方法を提供することである。

本発明の他の目的は、縮小命令集合信号プロセッサのアーキテクチュアがハードウェア（"シリコン"）に設計され且つ縮小され得ると共に（縮小命令のなかの）複合命令集合のコードエミュレーションのためのプログラミングが同時に行われる信号プロセッサを迅速に設計するための方法を提供することである。換言すれば、本発明のこの目的は、信号プロセッサが迅速な設計サイクルを有する複合命令集合をエミュレートするための縮小命令集合信号プロセッサを提供することである。

本発明の第一の局面によれば、ハーバード・アーキテクチュアを有する、すなわち別々のデータ及び命令バスを有する縮小命令集合計算機（ＲＩＳＣ）が提供される。

本発明の第二の局面によれば、ＲＩＳＣが複合命令集合計算機（ＣＩＳＣ）のエミュレータとして使用するため設計されている。ＲＩＳＣはそのオペランド又はデータバスを経て受信された複合命令に、ＲＩＳＣ命令の各々複合命令の一つに対応する複数個の群の対応する一つを、ＲＩＳＣ命令バスを経て、アドレス指定することにより、応答する。いったん受信された複合命令に対応する群の第一の命令がアドレス指定され且つ実行されると、特定の群のなかの組み合わされている縮小命令の残余は、群のなかの命令の全てが完全に実行され終わるまで、順次にアドレス指定され且つ実行され得る。こうして、複合命令が縮小命令の群を使用してハードウェアのなかでエミュレートされる。縮小命令はＲＩＳＣ機械の命令バスを経て受信される。

さらに本発明のこの第二の局面によれば、ハーバード・アーキテクチュアを有する縮小命令集合計算機が、設計される。複合命令集合計算機をエミュレートするための縮小命令集合信号プロセッサとしてだけでなく、以前のようにＲＩＳＣ命令

に対する命令バスを使用して但しこのモードではオペランドに対してのみデータバスを使用して縮小命令集合作動モードで作動する縮小命令集合信号プロセッサとしても使用するために設計される。これはＲＩＳＣ作動モードとして特徴付けられ得る。希望によりＲＩＳＣ又は複合命令エミュレーションモードが選択され得る。

さらに本発明の第二の局面によれば、縮小命令の複数個の群が信号プロセッサの外部のメモリストアのなかに記憶される。上記のように、群のなかの命令は実行のために順次に記憶される。換言すれば、群のなかの第一の縮小命令のアドレス指定は群のなかの他の縮小命令の各々の順次の実行により迅速に追随される。

さらに本発明の第二の局面によれば、エミュレートされる複合命令集合はＭＩＬ－ＳＴＤ－１７５０命令集合である。

本発明の第三の局面によれば、複合命令集合命令に応答する信号プロセッサの設計方法は、第一に、縮小命令集合信号プロセッサとして使用するため別々のデータ及び命令バスを有する縮小命令集合信号プロセッサを設計する過程と、第二に、複合命令集合をエミュレートするべく縮小命令集合信号プロセッサを設計する過程とを含んでおり、それによりデータバスが到来する複合命令集合命令及びそれに関係する双方向データの双方に対して使用される。ＲＩＳＣ命令バスは縮小命令集合命令のみをアドレス指定且つ受信するために使用される。複数個の縮小命令集合命令が受信された各到来複合命令に対して実行される。こうして、ハードウェアが、エミュレーションコードが同時に書かれるている間に実行するべく設計且つ縮小され得る。

さらに本発明の第一の局面によれば、信号プロセッサアーキテクチュアは、第一の入力信号及び第二の入力信号に論理演算を実行するため、また前記論理演算の結果を示すＡＬＵ出力信号を供給するため第一の入力信号及び第二の入力信号に応答する算術論理演算装置（ＡＬＵ）を有する。信号プロセッサアーキテクチュアは、ＡＬＵ出力信号を記憶し且つ"Ｂ"マルチプレクサ（ＢＭＵＸ）に供給するためＡＬＵ出力信号に応答するアキュムレータをも有する。"Ｂ"マルチプレクサは第二のＡＬＵ入力信号を供給するため、アキュムレータ出力信号を含む三つの異なる信号のなかから一つを選択する。ＢＭＵＸへの入力の他の一つは、ＡＬＵにより実行又は走査されるべきアドレスを成すＲＩＳＣ命令カウンタ出力信号である。ＢＭＵＸへの第三の入力は、レジスタファイルのなかの多数のレジスタの任意の一つに応答する宛先レジスタの出力である。宛先レジスタ出力信号が代わりにオペランドバス上にデータとして供給され得る。命令カウンタはＡＬＵから始動命令アドレスを受信し、またアドレスを順次にインクレメントするため規則的にクロックされ得る。ＡＬＵ出力信号はデータ又はオペランドバスからの到来オペランド信号に応答する多数のレジスタを含むレジスタファイルにも供給される。レジスタファイルは到来命令から復号されたものとして選択された記憶レジスタのなかにＡＬＵ出力信号もしくはオペランド信号を記憶する。レジスタファイルは宛先レジスタ及び源レジスタにレジスタを供給する。源レジスタはレジスタファイルから受信された信号を記憶し、またそれを"Ａ"マルチプレクサ（ＡＭＵＸ）に又は出力としてオペランドアドレスバスに供給する。ＡＭＵＸはＲＩＳＣ命令カウンタにより先にアドレス指定されたＲＩＳＣ命令バスから受信されたＲＩＳＣ命令にも応答する。ＡＭＵＸはＡＬＵに第一の入力信号を供給する。ハード配線による制御装置は到来する複合命令を復号し、また適当な順序で上記のアーキテクチュアを作動させるために必要な制御信号の全てを供給する。

ハード配線による制御装置は、（ｉ）各機械サイクルの第一の四分の一の間にＲＩＳＣ命令カウンタにインクレメントする信号を供給することにより順次に記憶された命令をアドレス指定するため、（ii）選択された機械サイクルの第二の四分の一の間に、ＡＬＵにより演算されるそれらの信号内容を有するようにレジスタファイルのなかの

記憶レジスタを選択するため、また源及び宛先レジスタのなかに記憶するため選択されたレジスタ信号内容を第二の四分の一の間に供給するため、レジスタファイルに制御信号を供給するため、（iii）選択された機械サイクルの第二の四分の一の間に、メモリからレジスタファイルへ又はレジスタファイルからメモリへデータをロード又は記憶するべく命令に応答してオペランドアドレス及びデータパスをイネーブルするためのイネーブル信号を供給するため、（iv）選択された機械サイクルの第三の四分の一の間に開始して、ＡＬＵに対する前記第一の入力信号として供給するため源出力信号と命令信号との間を選択するためＡＭＵＸに第一の選択信号を供給するため、（v）選択された機械サイクルの第三の四分の一の間に開始して、ＡＬＵに対する第二の入力信号として供給するため宛先出力信号とアキュムレータ出力信号とと命令アドレス信号との間を選択するためＢＭＵＸに第二の選択信号を供給するため、（vi）選択された機械サイクルの第二の四分の一の間に開始して、ＡＬＵにＡＬＵ演算選択信号を供給することによりＡＬＵにより実行されるべき演算を選択するため、（vii）選択された機械サイクルの第一の四分の一の間に開始して、適当なレジスタにＡＬＵ出力選択信号を供給することによりレジスタファイル、アキュムレータ又は命令カウンタのなかにＡＬＵ出力信号を記憶するため、（viii）選択された機械サイクルの延長された第四の四分の一の間に、シフト、乗算及び除算を実行するためシフト信号を供給するため命令信号を復号する。

さらに本発明の第二の局面によれば、かかるＲＩＳＣ信号プログラムはさらに、順次に記憶された縮小命令集合信号の群の第一の命令信号をアドレス指定するため縮小命令集合アドレス信号を復号し且つＲＩＳＣ命令カウンタに供給するためかかる複合命令信号を記憶し且つ制御手段に供給するためデータパスを経て受信される複合命令で書かれた応用プログラムに関係して供給される複合命令に応答する複合命令集合プログラムレジスタを含んでいる。アドレス指定されるかかる群の各々は、それぞれ複合命令集合のなかの複合命令の一つをエミュレートするために設計されている複数個のかかる群の一つである。複合命令集合プログラムアドレスカウンタは、順次に応用プログラムのなかの次の複合命令をアドレス指定するため、インクレメントする信号に応答し、又は例外複合命令アドレス信号を記憶及び供給するためＡＬＵ出力信号に応答する。

さらに本発明の第二の局面によれば、受信された複合命令信号は制御手段により復号され、また選択された機械サイクルの第一の四分の一の間にインクレメントする信号をプログラムカウンタに供給することにより順次にアドレス指定される。制御手段は選択された機械サイクルの第三の四分の一の間にＲＩＳＣ命令アドレスカウンタへのかかる縮小命令の群を開始するため縮小命令集合アドレス信号をも供給する。また制御手段は複合命令信号を受信、記憶及び供給するべく選択された機械サイクルの第一の四分の一の間に複合プログラムレジスタをイネーブルする。

本発明の第一の局面によれば、ＣＩＳＣのエミュレーションをなんら参照することなしに、ＲＩＳＣ機械として簡単に設計及び使用され得る簡単なアーキテクチュアが提供される。もちろん、背景技術のところで説明したように、かかる簡単なアーキテクチュアは同時にＲＩＳＣ属性及びＣＩＳＣ属性を有し得るので、二つの間のハードな速い境界を作ることは困難である。従って、本アーキテクチュアは二つの極限の間の設計空間に置かれ、上記の広い"縮小命令"の意味で例外として実際に厳密に特徴付け可能でないことは理解されよう。本明細書中に開示される"縮小"命令集合は、ここに開示されるアーキテクチュアで実行するために特に良く適している。しかし、他のＲＩＳＣ命令集合がここに開示されるアーキテクチュアでの実行のために定型化され得ることは理解されよう。

それにもかかわらず、提供される簡単なアーキテクチュアは、制限なしに、以下に説明されるいくつかの異なるＲＩＳＣアーキテクチュア・アプ

ローチをとる。第一に、シフト、乗算及び除算命令を除く全ての命令は単一の"２クロック"機械サイクルのなかで実行される。オペランドはロード及び記憶アクセスメモリのみを有するレジスタ・ツー・レジスタである。これは内部制御を簡単化する。全てのＲＩＳＣ命令はハード配線による制御を使用して実行される。マイクロコードは使用されない。３２命令演算のみが実行される。アドレス指定はレジスタ間接及び即値モードに制限されている。命令フォーマットは簡単であり、また境界を横切らない。

ここに開示されるＲＩＳＣアーキテクチュアに加えて、信号プロセッサが複合命令集合から受信された複合命令に実際に応答する意味でＣＩＳＣがエミュレートされ得る、本発明の第三の局面による、特に有用な設計方法が開示される。ＲＩＳＣ機械により受信されたこのような各複合命令は、エミュレーションによりこのような各命令を実行するため予めエミュレートされた縮小命令の特定の群のアドレス指定をトリガする。こうして、ドウェアが同時に縮小されるので、迅速なターンアラウンドの利点を提供する。しかし、速度は唯一の利点ではない。いったんハードウェア設計が"シリコン"に縮小されると、設計プロセスはそれに関しては完了され、その後の変更はなされ得ない。設計エラーはたいていこの段階では必然的に発見され、他の費用のかかる設計サイクルを必要とし、第二の設計の後にも追加的なエラーが発見される可能性がある。これは時間及び金の点で非常に費用のかかるプロセスである。本発明は、ハードウェア設計"問題"を受けるけれども、それにもかかわらずハードウェア問題を迂回するべくエミュレーションコードを使用する補正を受け得るハードウェアのなかに非常に簡単なアーキテクチュアを設計することにより、この問題の回避を許す。当業者に知られているように、設計プロセスの間に、機能は通常は一つよりも多い仕方で、所与の信号プロセッサのなかで、設計者により影響され得る。こうして、もしＲＩＳＣ機械を使用してＣＩＳＣをエミュレートすることを試みるならば、ＲＩＳＣを設計し、また、最初の設計のなかで最初に意図された仕方で複合命令が実行されるのを許さないハードウェア設計のなかの"欠陥"が存在する場合には、設計者は他の仕方で"問題"複合命令をエミュレートするべくエミュレーションコードのまわりを変更し得る。こうして、本アプローチはハードウェア設計欠陥を許容するべくフレキシビリティを提供する。

ここに開示される、ＲＩＳＣを使用する複合命令集合計算機を実現するための設計方法はハードウェアとソフトウェアとの間にＣＩＳＣを設計するタスクを分割する。設計効率はソフトウェアの効率から来る。従来の技術では、全ての機能は一つのチップ上にあった。本開示は、制御機能が実行機能から分離されている"２チップ解決"の使用を教示する。ここに開示される簡単な機械はハードウェア設計プロセスを速くし、またエミュレーションコードの同時設計を許す。従来の技術で経験された設計リスク、すなわち正しく作動する設計を得る以前にいくつかのハードウェア繰り返エミュレートされた各複合命令に対して、ハードウェア内での呼出し及び実行のために記憶された縮小命令の群が存在する。この設計アプローチはＲＩＳＣアーキテクチュアがハードウェア実施例に対して迅速に、すなわち同時にエミュレーションコードが開発されている間にゲートアレーとして設計されることを許す。こうして、もし設計者がＣＩＳＣの設計を望むならば、設計者は、同時に複合命令集合をエミュレートするためエミュレーションコードの開発を続行しつつ、先ずＲＩＳＣを設計し、続いてそれを"シリコン"に縮小することにより目的物を一層迅速に得ることができる。このアプローチの費用は純粋なＣＩＳＣとしてＣＩＳＣを設計する費用の１０％よりも少なく、且つ優れた結果が得られると見積もられる。主な費用節減の一つは、いくつかの設計サイクルの形態でＣＩＳＣ設計者により通常経験される設計リスクに関係付けられる。

この設計アプローチは、同時にエミュレーションコードが書かれている間に"シリコン"にハー

しにさらされるリスクは回避される。

本発明の第二の局面は、ＣＩＳＣをエミュレートするため群のなかに配置されたＲＩＳＣコードを実行するＲＩＳＣ機械を提供する。この局面の実施例は、純粋なＲＩＳＣモードでハーバード・アーキテクチュアを有する純粋なＲＩＳＣ機械として、もしくはＣＩＳＣをエミュレートするため群のなかに配置されたＲＩＳＣコードを実行するＲＩＳＣ機械として、ＲＩＳＣ機械をランさせる可能性を提供し得る。いずれの場合にも、コードのこのような各群は、対応する複合命令に応答して実行され得る。いったん群の初期アドレスがＲＩＳＣアドレスバス上に置かれると、その群のなかの縮小命令の他のメンバーが順次に実行される。ＲＩＳＣコードの群によりエミュレートされるＣＩＳＣを有する新規なアプローチと、まさにＲＩＳＣとして又はエミュレータとしてランするＲＩＳＣ機械を有する新規なアプローチとは縮小命令集合のフィロソフィによる強力な信号処理ツールを提供し、また縮小命令集合アーキテクチュアを使用する複合命令集合をエミュレートする可能性を同時に提供する。本発明のこの局面は、こうして使用される時、まさに強力なツールである。

ＲＩＳＣコードのなかで実行されるべきユーザー固有の、ユーザーにより定められた機能をユーザーが開発し得ることは、本発明の第二の局面により教示される縮小命令集合信号プロセッサの他の強力な特徴である。通常、ＭＩＬ－ＳＴＤ－１７５０によるＣＩＳＣの製造者は、もしその顧客がそのＭＩＬにより許されるようなユーザーにより定められた機能を実現することを望むならば、その顧客が特注の特別に開発されたＭＩＬ－ＳＴＤ－１７５０によるマイクロプロセッサチップを注文することを要求する。（ＭＩＬ－ＳＴＤ－１７５０Ａはユーザーにより定められた機能を暗示的に許し、他方に於いてＭＩＬ－ＳＴＤ－１７５０Ｂはこのような機能を明示的に準備する）。本発明はユーザーに、チップが購入された後にＲＩＳＣコードのなかでの実行のために任意の数のこのようなユーザーにより定められた複合命令を定める可能性を与える。これらは、ＲＩＳＣ機械と組み合わされており、またＲＩＳＣマイクロプロセッサと共に使用するためのＰＲＯＭのようなメモリ装置のなかに別々に記憶されているＲＩＳＣ命令集合を使用して、簡単に顧客により設計され得る。完全なＲＩＳＣ命令集合はもちろん、ユーザーにより定められた機能を顧客が書くことを可能にするべくＭＩＬ－ＳＴＤ－１７５０マイクロプロセッサの購入時に供給される製品文献のなかで顧客に説明されなければならない。

本発明の第一の局面の以上の教示の全てに追加して、特殊な縮小命令集合が、前記のように、その命令集合及び多くの他の可能な類似の命令集合を実現するために特に有用である特殊なＲＩＳＣアーキテクチュアとならんで教示される。開示される特殊なＲＩＳＣアーキテクチュアは後で開示されるような第二の局面によりＭＩＬ－ＳＴＤ－１７５０命令をエミュレートする目的にも非常に有用である。

本発明の前記及び他の目的、特徴及び利点は以下にその好ましい実施例を図面により詳細に説明するなかで一層明らかになろう。

発明を実施するための最良の形態

第１図には、縮小命令集合計算機（ＲＩＳＣ）として広く特徴付けられてよく、また通常は、制限なしに、マイクロプロセッサの形態をとる、本発明の第一の局面による簡単な信号プロセッサ１０がブロック図で示されている。本発明の新規なＲＩＳＣはハーバード・アーキテクチュアを有する。すなわち命令及びデータバスが別々である。ＲＩＳＣ命令アドレスバス１２は縮小命令をアドレス指定するのに使用されており、縮小命令は次いでＲＩＳＣ１０により縮小命令バス１４を経て受信される。オペランドアドレスバス１６は双方向であってよいデータバス１８を経てＲＩＳＣ１０により受信もしくは供給されるデータをアドレス指定するのに使用される。背景技術のところに記載したように、頭辞語“ＥＩＳＣ”はここでは縮小命令集合機械の広い意味で使用されている。

ハーバード・アーキテクチュアの使用は、ノイ

マン・アーキテクチュアが好ましいアプローチであった以前のＣＩＳＣに使用されているような多重化命令／データパスとは異なっている。ハーバード・アーキテクチュアによるアプローチは、命令及びデータが同時にアクセスされ得るという事実により、ＲＩＳＣがはるかに速い速度で作動することを許す。これは性能を最大化する上記の一般的ＲＩＳＣアプローチと調和している。

第２図は第１図のＲＩＳＣと類似しておりまた同一のハーバード・アーキテクチュアを有するＲＩＳＣ１０のブロック図である。第２図のＲＩＳＣ１０は、共に使用される時に、本発明の第二の局面に従って、複合命令集合計算機（ＣＩＳＣ）をエミュレートし得るプログララマブル・リードオンリメモリ（ＰＲＯＭ）のようなメモリ装置２０と組み合わされている。ＲＩＳＣ１０ＰＲＯＭ２０は共にＣＩＳＣをエミュレートするべくＲＩＳＣを使用する構想の＂２チップ＂２２の実施例又はそれへの＂解決＂として考えられ得る。２チップ２２は実際にＣＩＳＣである。これは設計者及びユーザーの双方の観点から上記のように高度に効率的なアプローチである。

信号の形態の複合命令は信号線２４を経てデータパスに供給され、それによりＲＩＳＣ１０に伝達される。複合命令２４の源は本発明にとって重要ではないが、一般に＂ＣＩＳＣ＂２２により実行されるべきいくつかの応用プログラムから成っている。複合命令はこの仕方で順次に受信され、またＲＩＳＣ１０に復号される。受信されるこのような各複合命令に対して、初期始動アドレスがＲＩＳＣ命令アドレスバス１２を経てＰＲＯＭ２０に供給され、そこにエミュレートされている特定のＣＩＳＣ命令に対応するＲＩＳＣ命令の群が順次に記憶される。ＰＲＯＭ２０のなかでアドレス指定された第一のＲＩＳＣ命令は次いで、典型的に全機械サイクルの間、ＲＩＳＣ命令バス１４上に置かれる。命令が記憶又は実行のためにＲＩＳＣ１０に安全に供給され終わった後、ＲＩＳＣ１０は次に、命令バス１４上に置くため次に順次に記憶されたエミュレーション命令をアクセスするためＲＩＳＣ命令アドレスバス１２上のアドレスをインクレメントする。このプロセスは、（データバス１０を経て受信された特定のＣＩＳＣ命令をエミュレートするためＰＲＯＭ２０のなかに記憶されている）ＲＩＳＣ命令の全ての群がエミュレーションを経由して実行され終わるまで継続する。所与の命令は、なかんずく、データバス１８及びその（単にデータ源又は宛先への経路を示す）延長部２６を経由してオペランドをロード又は記憶するのに使用されるメモリロード／記憶操作を含んでいる。

ＲＩＳＣ

第３図には、第１図及び第２図中に示されているＲＩＳＣに類似のＲＩＳＣ１０の内部の簡単化されたブロック図が示されている。第３図の細線は、本発明のＣＩＳＣエミュレーションの局面なしに、第１図のＲＩＳＣ１０に一致する。先ず、本発明の第一の局面のＲＩＳＣ実施例の動作を完全に説明するため、第３図を先ずこれらの細線の部分に関して説明する。次いで、本発明の第二の局面の実施例、すなわち第２図に示されているようにＣＩＳＣをエミュレートするのに使用されるＲＩＳＣを完全に説明するため、太線の部分を説明する。

制御装置３０はＲＩＳＣ１０に対する（図面を簡単にするため図示されていない）内部制御信号を供給する。このような制御信号により制御される機能のなかで次のＲＩＳＣ命令の取出しが実行されなければならない。これは、ＲＩＳＣ命令アドレスを保持するのに使用される命令カウンタ３２を、そのアドレスに記憶されているＲＩＳＣ命令を命令レジスタ３４のなかへロードするため、インクレメントすることにより成就される。命令レジスタ３４は実際には二つのレジスタ、すなわち（現在の機械サイクルの間に）命令バス１４上でＲＩＳＣ命令を最初に受信及び記憶するための＂パイプ＂又は一次レジスタと、現在の機械サイクルの中央まで先の機械サイクルの間に命令バス１４上にあった命令を保持するための二次レジスタとから成っている。パイプはその内容を、後で

一層詳細に説明するように、各機械サイクルの中央で二次レジスタに伝達する。

次に第４図を参照すると、制御装置３０により与えられる種々の制御信号に相当する波形が示されている。図示の目的で、第４図の種々の波形は四分の一機械サイクル４０、４２、４４、４６に分割されている単一機械サイクル３８のなかで相互に比較するため時間的に"凍結"されている。任意の選択された周波数又はデューティサイクルであってよい基本システム"クロック"波形は第４図（ａ）に示されている。第４図（ｂ）には後で一層詳細に説明する目的で有用である第一の四分の一４０のクロック信号波形５０（ＣＫ１）が示されている。第４図（ｃ）には各機械サイクル第二の四分の一４２の間に再び生ずる（この場合、高い）波形５２（ＣＫ２）が示されている。第４図（ｄ）には各機械サイクル第三の四分の一４４の間に再び生ずる波形５４（ＣＫ３）が示されている。第４図（ｅ）には各機械サイクル第四の四分の一４６の間に再び生ずる波形５６（ＣＫ４）が示されている。

第４図（ｆ）には、命令レジスタ３４のなかの"パイプ"レジスタのローディングのタイミングを示す波形５８が示されている。波形５８は、第４図（ｂ）の波形５０のＣＫ１の立ち上がり縁と同時に生起する低レベルから高レベルへの電圧変化の立ち上がり縁に小さい矢印を付して示されている。これは、第３図のＲＩＳＣ命令線１４上に存在する（全機械サイクルの間、その時点に存在するであろう）命令がいま安定しており命令レジスタ３４のなかのＲＩＳＣ命令レジスタパイプのなかへのクロッキングの準備ができていることを示す。こうして、第３図中に示されているクロック線６０は各機械サイクルの開始時にＣＫ１の立ち上がり縁に上がる。換言すれば、命令レジスタパイプの内容が各機械サイクルの開始時に変更される。命令が二つの全四分の一サイクルにわたり"パイプ"レジスタのなかに存在した後、命令は第４図（ｇ）中の波形６４の立ち上がり縁６２により示されているようなＣＫ３の開始時に"パイプ"から二次命令レジスタのなかへラッチされる。もちろん、第二の四分の一４２の間に二次命令レジスタの入力端に与えられる信号が、クロックが低レベルであるので、この第二の四分の一４２の間に二次命令レジスタの出力端にも存在していることは理解されよう。従って、まだラッチされていなとしても、この第二の四分の一周期の間に制御装置による命令コードの復号が開始され得る。こうして、"パイプ"のなかの命令は機械サイクルの開始時から終了時までそのなかに留まっているけれども、パイプの内容は各機械サイクルの中央で二次命令レジスタのなかへコピーされる。従って、所与の命令が二次レジスタのなかに存在する時間は一つの機械サイクルの中央から他の機械サイクルの中央まで重なっている。

命令バス１４上の命令が命令レジスタ３４"パイプ"のなかへクロックされる時点で、命令アドレスバス１２上のアドレスが、各機械サイクルの開始時にＣＫ１の立ち上がり縁と一致するその作動クロッキング（立ち上がり縁７０）クイミングを有する第４図（ｈ）の波形６８に示されているクイミングの制御信号により開始されて次の順次アドレスへ通常インクレメントされる。しかし、もし例外命令が実行されるべきであれば、それは順次アドレスの外でロードされる。

制御装置３０の他の機能は、種々の仕方で、命令レジスタ３４"パイプ"内及び命令レジスタ３４の二次レジスタ内に留まっている命令を復号することである。各機械サイクルの第一の四分の一の間に、制御装置３０は命令レジスタ３４のパイプレジスタのなかへまさにロードされた命令のレジスタフィールド部分を復号する。線７２はレジスタフィールドを復号するためのパイプから制御装置３０への情報の転送を示す。（ＲＩＳＣ命令フォーマットは後で一層詳細に説明する）。ここでは、示されている実施例に対して、１６ビットＲＩＳＣレジスタ・ツー・レジスタ命令フォーマットに対して、６個の最上位ビットが命令コードであり、次の５個の最上位ビットが宛先フィールドであり、また最後の５個のビットが源フィール

ドであると言えば十分である。復号されたレジスタフィールド情報は、パイプのなかで、宛先レジスタ７６及び源レジスタ７８のなかへロードされるべきレジスタファイル７４のなかのレジスタの選択された対を同定するのに使用され得る。いったんレジスタフィールドが復号されると、機械サイクルの第二の四分の一４２の間に能動化される選択線８０により示されているように、レジスタの適当な対が選択される。第４図（ｉ）及び第４図（ｊ）には、それぞれの立ち上がり縁８６、８８により、レジスタファイル７４のなかの選択されたレジスタがその時点で、すなわち機械サイクルの中央で宛先及び源レジスタ７６、７８のなかへクロックされることを示す波形８２、８４が示されている。

この時点で、上記のように、パイプのなかの命令が命令レジスタ３４のなかの二次レジスタのなかへラッチされる。命令レジスタの二次レジスタのなかで命令コードが、実行されるべき命令を決定する目的で復号される。宛先フィールドもこの時点で再び、内部演算の結果もしくはデータバス１８から移動されたオペランドにより書かれるべきレジスタファイル７４のなかのレジスタを決定する目的で復号される。これらの二つの代替的なローディング経路は信号線１００及び信号線１０２により示されている。信号線１００はなかんずく宛先及び源レジスタ７６、７８により与えられる信号を演算する算術論理演算装置（ＡＬＵ）１０４の３２ビット出力を表す。宛先及び源レジスタ７６、７８の内容がＡＬＵ１０４により演算される演算に対して、その演算の出力は線１００を経て、宛先レジスタを最初にロードしたレジスタ対のなかへ戻される。他方、情報はバッファ１０６、信号線１０８、入力／出力（Ｉ／Ｏ）装置１１０及び信号線１０２を経てデータバス１８から与えられるものであってもよい。Ｉ／Ｏ装置は、これまでに説明した目的で、簡単に信号線１０８から信号線１０２への短い回路と考えられてよい。すなわち、それは本開示の現在のレベルに重要な機能はしない。（しかし、本発明のＭＩＬ－ＳＴＤ－１７５０実施例と結び付いて本開示の後続のレベルに関しては重要であり、従ってその目的で第３図に含められている）。

もし制御装置３０が、命令レジスタ３４のなかの二次レジスタから復号されたものとして、レジスタファイル７４のなかのレジスタ又はレジスタ対のなかへ信号線１００上のＡＬＵ出力信号を含むＡＬＵ１０４に対する演算を選択すれば、ＡＬＵの出力は第４図（ｋ）の波形１１４の立ち上がり縁１１２により示されているように機械サイクルの第二の四分の一の開始時にレジスタファイル７４のなかの選択されたレジスタ又はレジスタ対のなかへクロックされる。この波形はＡＬＵ１０４により書込まれるべきレジスタをクロックするため第３図中に示されているクロック線１１６上に存在する電圧を表す。

こうして、各機械サイクルの第一の四分の一４２の間に、制御装置３０は宛先及び源レジスタ７６、７８をロードするためアクセスされるべきレジスタファイルのなかのレジスタの対を決定するべく命令レジスタ３４パイプを復号し、またレジスタファイル７４のなかのどのレジスタが入力バスで又は以前の機械サイクルからのＡＬＵの出力でロードされるべきかを決定する目的で命令レジスタ３４のなかの二次レジスタを復号する。線１００上のＡＬＵ出力信号又は線１０２上のデータはＣＫ２の立ち上がり縁で選択されたレジスタ又はレジスタ対のなかへロードされ、他方宛先及び源レジスタはＣＫ３の立ち上がり縁でロードされる。

もし命令レジスタ３４のなかの二次レジスタのなかの命令から復号された命令コードが外部メモリ空間からデータをロードするため又はそれにデータを記憶するためのロード／記憶命令であることが判明すると、信号線１１８及びバッファ１２０を経て源レジスタ７８からオペランドアドレスバス１６上へオペランドアドレス出力が与えられなければならない。制御装置３０はもちろん、線１１８上の信号をオペランドアドレスバス１６に通す目的で、バッファ１２０をイネーブルする。

これは信号線１１７ａ及び１１７ｂ上でクロックされるＣＫ３の立ち上がり縁で生起する。ロード操作時に、オペランドは次いでメモリからデータバス１８上に現れ、制御装置３０によりイネーブルされたバッファ１０６と線１０８とＩ／Ｏ１１０と信号線１０２とを経てレジスタファイル７４のなかへ送られる。命令レジスタ３４のなかの二次レジスタのなかに留まっているＲＩＳＣ命令の宛先フィールドは、それがＡＬＵ演算に関する命令に対するものであったかのように、オペランドがロードされるべきレジスタファイル７４のなかのレジスタを指定するために使用される。再び、これは第４図（ｋ）の波形１１４に示されているようにＣＫ２の立ち上がり縁で生起する。記憶操作時に、オペランドは、宛先レジスタ７６のなかへロードされる命令の宛先フィールドにより指示されたレジスタからデータバス１８上に供給される。

ここまでは、制御装置３０は、命令カウンタ３２をインクレメントすることによりＲＩＳＣ命令を取出し、レジスタファイル７４のなかで演算されるべきレジスタを選択し且つそれらを源及び宛先レジスタ７６、７８のなかへ記憶することを含めて、ＲＩＳＣ機械のなかで種々の機能を実行するため命令レジスタ３４の一次（"パイプ"）レジスタ及び二次レジスタの双方のなかに受信された現在の命令を復号し、またレジスタファイル７４のなかの選択されたレジスタのなかへ先の命令からのＡＬＵ出力信号をロードし、もしくはもし復号されたオペランドがメモリからのロード操作を示すならばデータバス１８からのオペランドをロードするその機能を実行するものとして説明されてきた。

説明されるべき次の機能は、マルチプレクサ１２４、１２６の対を通じてのＡＬＵ１０４への入力の選択である。第一のマルチプレクサ１２４は第一の入力信号１２８をＡＬＵ１０４の第一の入力端に与える。第二のマルチプレクサ１２６は線１３０上の第二の入力信号１２８をＡＬＵ１０４の第二の入力端に与える。これらの入力信号は各マルチプレクサへのいくつかの異なる入力信号のなかから選択線１３２、１３４により選択される。選択線は二次命令レジスタを復号することにより用意され、またＣＫ３の立ち上がり縁の生起時に存在している。換言すれば、宛先及び源レジスタ７６、７８が選択されたレジスタでロードされる時、マルチプレクサが選択される。もし宛先及び源レジスタ７６、７８がそれらの内容をＡＬＵ１０４により演算するために選択されるならば、それらの内容が適当な演算を実行するため線１１８及び１３５を経てまたそれぞれのマルチプレクサ１２４、１２６を通じてＡＬＵの適当な入力端に転送される。制御装置３０は命令コードが二次命令レジスタのなかに留まる間に命令コードを復号することによりＡＬＵにより実行されるべき演算を選択する。これは通常ＣＫ３の立ち上がり縁の少し後に生起する。ＡＬＵは加算、アンド、オア及び排他的オアを含む標準レパートリの演算を行い得る。いったんマルチプレクサの出力が安定化すると、演算は安定な出力を線１００上に与えるべくハードウェア内で安定化される。

第３図中に示されているＲＩＳＣ１０に対して、本開示の現在のレベルで、すなわち本発明の第一の局面と結び付いて、マルチプレクサ１２４、１２６の各々が唯二つの別々の信号に応答する。追加的な信号は本発明の第二の局面を説明するため本開示の他のレベルと結び付いて後で説明される。マルチプレクサの他の入力は、開示のＲＩＳＣレベルに対して、ＲＩＳＣ命令及び命令アドレスバスと結び付けられている。こうして、マルチプレクサ１２４は制御装置３０からの線１２６上の命令信号に応答する。これは、ＲＩＳＣ命令上で演算を実行することが望まれた特定の機械サイクルの第三の四分の一４４の間にマルチプレクサ１２４のなかへ転送される命令レジスタ３４の二次レジスタのなかに留まっている即値データフィールドに一致する。しかし、これらは例えば、（命令のなかの）即値データフィールドよりもむしろデータが操作される通常の内部演算に比較して大きな間隔をおいている。レジスタ・ツー・即値

命令は後でＲＩＳＣ命令集合と結び付けて説明される。マルチプレクサ１２６は、開示のこのレベルで、命令アドレスバス１２からのＥＩＳＣ命令アドレス信号にも応答する。このようなアドレスは時々操作され得る。こうして、制御装置３０は、命令コードに従って、ＡＬＵのなかでの演算のためにマルチプレクサ１２４、１２６を通じてＡＬＵへの入力を選択する。制御装置３０は、命令コードに従って、ＡＬＵにより入力信号上で実行されるべき特定の演算をも選択する。演算は、入力が安定になった後に線１００上に安定な出力信号を生ずるハード配線による論理回路により実行される。これは通常、機械サイクルの後半の間及び次の機械サイクルの第一の四分の一の間に行われる。ＡＬＵの出力は、第４図（ｋ）の波形１１４の立ち上がり縁１１２に相当する次の機械サイクルの第二の四分の一の立ち上がり縁まで意図される宛先のなかへロードされない。線１００上のＡＬＵ出力信号に対する宛先はレジスタファイル７４のなかのレジスタ、命令カウンタ３２又はアキュムレーク１４０であってよい。アキュムレークは、シフト、乗算及び除算を実行する目的で設けられており、また、ＡＬＵ演算の後でＡＬＵ出力信号をレジスタファイルにロードする場合のように、選択された機械サイクルの間にＣＫ２の立ち上がり縁でＡＬＵの出力をロードされる。

制御装置３０はシステム受信器を更新し、また例外プログラムフロー（割込み、呼出し、飛越し）をチェックする。適当な制御信号がこのような場合に第３図中に示されている種々の機能エンティティーに与えられる。

次に第５図を参照すると、第３図のレジスクファイル７４が一層詳細に示されている。内部データバス１０２は内部データバス１０２上の信号と線１００上のＡＬＵ出力信号との間を選択する第一の３：１マルチプレクサ１５０及び第二の３：１マルチプレクサ１５２に導かれている。線１００は、第５図では、ＡＬＵ出力信号の最上位の半分を導く信号線１００ａとその最下位の半分を導く信号線１００ｂとに分割されている。これらの線１００ａ及び１００ｂの各々は第一及び第二のマルチプレクサ１５０、１５２の双方に導かれている。第３図の制御装置３０は、どのマルチプレクサ１５０、１５２及びどの経路１００、１０２がレジスクに信号を与えるかを制御する。

レジスクファイル７４自体は２０個の汎用レジスタの群を含んでいる。レジスクファイルは信号線１００からのＡＬＵデータもしくは（データバス１８から出発する）信号線１０２からのオペランドデータをロードされる。ＲＩＳＣ命令はこれらのレジスタ上で演算を実行する。第５図には、二つの群に分割されたレジスタが示されており、第一の（"偶数"）群７４ａ（R_0～R_{14}＆Ａ０＆Ａ２）は第一のマルチプレクサ１５０からの線１５４上の出力信号に応答し、また１０個の汎用レジスタを含んでいる。汎用レジスタ７４ｂの第二の（"奇数"）群（R_1～R_{15}＆Ａ１＆Ａ３）は第二のマルチプレクサ１５２からの線１５６上の第二の信号に応答する。いずれかの群７４ａ、７４ｂのなかのレジスタのいずれかがその内容を信号線１６２、１６４を経て第三のマルチプレクサ１５８又は第四のマルチプレクサ１６０に供給し得る。第三のマルチプレクサ１５８は出力信号１６６を第３図の宛先レジスタ７６に供給するための２０：１マルチプレクサである。第四のマルチプレクサ１６０は線１６８上の出力信号を第３図の源レジスタ７８に供給するための２０：１マルチプレクサである。

第３図の実施例では、ＡＬＵは３２ビットＡＬＵであるが、データ及びデータアドレスバス１８、１６も命令及び命令アドレスバス１４、１２も１６ビットである。従って、第５図の構造に対しては、内部データバス１０２は１６ビットであり、ＡＬＵ信号出力線１００ａの最上位の半分の信号も線１００ｂ上の最下位の半分の信号も１６ビットである。従って、マルチプレクサ出力線１５４、１５６も汎用レジスタ７４ａ、７４ｂの全ても１６ビットである。２０：１マルチプレクサ１５８、１６０の各々は３２ビット語を構成し、最上位の半分は群７４ｂのなかの１０個のレジスタ

の一つから取られ、また最下位の半分は群７４ａもしくは７４ｂのなかのレジスタのいずれか一つから取られる。こうして、宛先及び源レジスタ出力信号１６６、１６８は３２ビット語である。

次に第６図を参照すると、第３図中に示されている源モジュール７８のような源モジュールがブロック図で示されている。線１６８上の信号は３２ビット信号であり、線１７６上の上位の１６ビット及び線１７８上の下位の半分としてそれぞれ第一の３：１マルチプレクサ１８０及び第二の３：１マルチプレクサ１８２に与えるため点１７４で二つの１６ビット語に分割されている。点１８４で線１６８上の３２ビット語の最下位の５ビットは線１８６を経て、全てのビット演算に対する適当なビットを選択するビットデコーダ１８８に与えられる（ビットは変更される）。ビットデコーダ１８８からの３２ビット語は、それぞれ第一及び第二のマルチプレクサ１８０、１８２に与えるため、線１９２上の最上位の半分と線１９４上の最下位の半分とに分割される。第一のマルチプレクサ１８０はレジスタファイルの最上位の半分の出力を選択し、又はビット命令上のビットフィールドを選択する。それはレジスタファイルのビット１６出力を符号拡張する（全て１又は０に強制する）。第二のマルチプレクサ１８２はレジスタファイルの最下位の半分の出力を選択し、又はビット命令上のビットフィールドを選択する。それはバイトスワップ命令上で下位の８ビット及び上位の８ビットをスワップさせる。

線１６８ａ上の選択された最上位の半分及び線１６８ｂ上の選択された最下位の半分は、第１図と結び付けて先に説明したようにＡＬＵ演算の実行前に導き出された源データフィールドを保持するのに使用される源レジスタ７８に与えられる。源はメモリへ又はそれからレジスタをロード又は記憶する以前にオペランドアドレスを与えるのにも使用される。これは第３図に示されており、出力線１１８がバッファ１２０に導かれており、バッファ１２０がオペランドアドレス信号を線１６を経てオペランドメモリ空間に与える。前記のように、出力信号１１８は、もしオペランドアドレス指定機能を実行しないならば、第３図に一層良く示されているように、ＡＬＵに第一の入力信号１２８を与えるための命令線１３６とならんで、マルチプレクサ１２４に与えられる。

次に第７図を参照すると、第３図中の宛先モジュール７６と類似の宛先モジュールがブロック図で示されている。レジスタファイル７４のなかのマルチプレクサ１５８からの線１６６上の出力信号は、１６及び３２ビット語上の各クロック信号上の右シフト、左シフト又はノーシフトを許す３２ビットのシフターマルチプレクサ２００に与えられる。それは論理的、算術的及び循環的にシフトする。シフターマルチプレクサ２００の出力は線２０２を経て宛先レジスタ２０４に与えられる。このレジスタはシフターマルチプレクサ２００の導き出された宛先データフィールド出力を保持するのに使用される一時レジスタとして考えられ得る。宛先レジスタ２０４の出力は線１３６を経て、線１３０上の第二の入力信号を第３図のＡＬＵ１０４に与えるためのマルチプレクサ１２６に与えられる。宛先モジュール７６はメモリにレジスタを記憶する以前にオペランドデータを与えるためにも使用されている。宛先モジュールはクロックあたり１ビットだけ右又は左にシフトされ得るし、またシフト、乗算及び除算機能を実行するのに使用されている。

次に第８図を参照すると、第３図中のアキュムレータ１４０と類似のアキュムレータモジュール２１０がブロック図で示されている。線１００上のＡＬＵ出力信号は、線２１８上のアキュムレークレジスタ２１６の出力信号に応答する２：１マルチプレクサ２１２に与えられる。このマルチプレクサは３２ビットの出力信号を線２２０を経て、乗算、除算及び６４ビットシフトのために必要とされるような右シフト、左シフト又はノーシフトを許すシフターマルチプレクサ２２２に与える（それは宛先レジスタと連結されている）。シフターマルチプレクサ２２２の出力は線２２４を経てアキュムレータレジスタ２１６に与えられる。

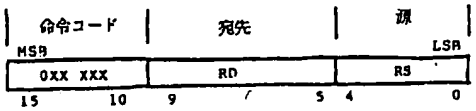
アキュムレータモジュール２１０は全体としてデータの一時記憶のためにも使用され得る。乗算及び除算の結果は宛先モジュール７６及びアキュムレータモジュール１４０又は２１０のなかで定型化される。

第３図の信号プロセッサは１６及び３２ビット命令を支持する二つの基本命令フォーマットを有する。命令コード（ｏｐｃｏｄｅ）は命令の６個の最上位ビットから成っている。第１表には３２命令の命令コードマトリックスが示されている。命令コードの上側の２ビット及び次に下側の３ビットはそれぞれ、命令が置かれている行及び列を選択する。二つの命令フォーマットは（１）レジスタ・ツー・レジスタ（ＲＲ）及び（２）レジスタ・ツー・即値（ＲＩ）である。６ビット命令コードの最下位ビットは、それにより命令が解釈されるべきフォーマットを選択する。３２命令の各々は第６ビットに関係して各フォーマットのなかで実行され得る。

第1表

	00	01	10	11
000	MOV	ADD	AND	SLL
001	LR	ADDC	OR	SAR
010	STR	AB	XOR	SCR
011	CALL	ADDU	NOT	MULS
100	MOVC	SUB	RBR	MOVB
101	INR	SUBB	SBR	SWAB
110	OTR	SB	TRR	DIV
111	JCR	CMP	LRI	STRI

レジスタ・ツー・レジスタフォーマットは６ビットの命令コード及び二つの５ビットのレジスタフィールドから成る１６ビットの命令である。レジスタフィールドは（１）２０個の汎用レジスタ、（２）１０個の汎用レジスタ対又は（３）１個のアキュムレータのいずれか一つを選択し得る。



レジスタ・ツー・レジスタ宛先（ＲＤ）及び源フィールド（ＲＳ）は第２表に従って選択され得る。

第2表

宛先及び源フィールド選択

ビットフィールド値	RDが選択する時	RSが選択する時	ビットフィールド値	RDが選択する時	RSが選択する時
00000	R0	R0	10000	XR0	XR0
00001	R1	R1	10001	XR2	XR2
00010	R2	R2	10010	XR4	XR4
00011	R3	R3	10011	XR6	XR6
00100	R4	R4	10100	XR8	XR8
00101	R5	R5	10101	XRA	XRA
00110	R6	R6	10110	XRC	XRC
00111	R7	R7	10111	XRE	XRE
01000	R8	R8	11000	A0	A0
01001	R9	R9	11001	A1	A1
01010	RA	RA	11010	A2	A2
01011	RB	RB	11011	A3	A3
01100	RC	RC	11100	XA0	XA0
01101	RD	RD	11101	XA2	XA2
01110	RE	RE	11110	---	*IMM
01111	RF	RF	11111	ACC	ACC

＊Ｒ＊は源に対して１６ビット拡張フィールドを使用する。

第３表には第５図のレジスタファイルの二つの半部に対して選択される組織が示されている。

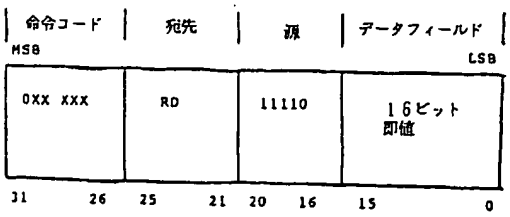
第3表

１６ビット	１６ビット	レジスタ対
R0	R1	XR0
R2	R3	XR2
R4	R5	XR4
R6	R7	XR6
R8	R9	XR8
RA	RB	XRA
RC	RD	XRC
RE	RF	XRE
A0	A1	XA0
A2	A3	XA2
ACC		

表の左側の（１６ビット）列は１０個のレジスタ７４ａに相当し、表の右側の（１６ビット）列は１０個のレジスタ７４ｂに相当する。前記のように、レジスタは３２ビット語を形成するため対として作動するように選択され得る。レジスタ対に対する命名は第３表に示されており、またレジスタ・ツー・レジスタ命令のなかのレジスタフィールドによる選択のために第２表中に反映されて

いる。A0～A3レジスタは一般に中間結果を保持するために使用される。

レジスタ即値フォーマットは6ビットの命令コードと一つの5ビットのレジスタアドレスと即値命令を示す5ビットのコードと16ビットのデータフィールドとから成る32ビットの命令である。



レジスタ／アキュムレータ・ツー・即値フォーマットは命令コードマトリックスの命令を全て包含する。レジスタ／アキュムレータフィールドは第2表中に示されているRDフィールドとして選択され得る。

RISC命令集合

アドレスモード	簡略記号	レジスタ転送記述	CPZNV	影響されるレジスタ
RR	MOV RD,RS	RD <-- RS	*****	RD

RR	LR RD,RS	IF (RS = SP)
		THEN
		RS <-- RS + 1;
		RD <-- @(RS);
		IF (RS != SP)
		THEN
		RS <-- @(RS);
			*****	RD,SP

RR	STR RD,RS	IF (RS = SP)
		THEN
		RS <-- RS - 1;
		RS <-- @(RD);
		IF (RS != SP)
		THEN
		RS <-- @(RD);
			*****	RS,SP

RR	CALL RD,RS	RD <-- PC + 2 PC <-- RS	*****	PC, SP
RR	MOVC RD,RS	RD <-- RS	0PZN0	RD, SW
RR	INR RD,RS	RD <-- @(RS);	*****	RD, SP
RR	OTR RD,RS	@(RS) <-- RD;	*****	SP

RR	JCR N,RS	IF (SW = N)
		THEN
		PC <-- PC + RS;
		ELSE
		PC <-- PC + 1;
			*****	PC

RR	ADD RD, RS	RD <-- RD + RS;	CPZNV	RD, SW
RR	ADDC RD,RS	RD <-- RD + RS + C;	CPZNV	RD, SW
RR	AB RD,RS	RD_{7-0} <-- RD_{7-0} + RS_{7-0};	CPZNV	RD, SW
RR	ADDU RD,RS	RD <-- RD + RS;	CPZN*	RD, SW
RR	SUB RD,RS	RD <-- RD + RS + 1;	CPZNV	RD, SW
RR	SUBB RD,RS	RD <-- RD + RS + C;	CPZNV	RD, SW
RR	SB RD,RS	RD_{7-0} <-- RD_{7-0} + RS_{7-0} +1;	CPZNV	RD, SW
RR	CMP RD, RS	RD : RS;	0PZN*	SW
RR	XOR RD,RS	RD <-- RD @ RS;	0PZN*	RD, SW
RR	NOT RD,RS	RD <-- RS;	0PZN*	RD, SW
RR	RBR RD,RS	RD <-- RD AND BIT(RS);	*****	RD
RR	SBR RD,RS	RD <-- RD AND BIT(RS);	*****	RD
RR	TBR RD,RS	ALU <-- RD AND BIT(RS);	0PZN*	SW
RR	LRI RD,RS	RD <-- (RS);	*****	RD
RR	SLR RD,RS	RD <-- RD SHIFT (RS);	0PZN*	RD, SW
RR	SAR RD,RS	RD <-- RD SHIFT (RS);	0PZN*	RD, SW
RR	SCR RD,RS	RD <-- RD SHIFT (RS);	0PZN*	RD, SW
RR	MULS RD,RS	RD <-- RD * RS;	****V	RD, SW
RR	MOVB RD,RS	RD_{7-0} <-- RS_{7-0};	*****	RD
RR	SWAB RD,RS	RD_{15-8} <-- RS_{7-0}; RD_{7-0} <-- RS_{15-8};	*****	RD
RR	DIV RD,RS	RD <-- RD / RS;	****V	RD
RR	STRI RD,RS	@(RS) <-- (RD);	*****	none

命令集合の詳細な説明

以下に説明されるRISC命令の各々はレジスタ・ツー・レジスタフォーマットに対して説明される。しかし、各々が命令コードの第6又は最下位ビットの状態により指定され得るレジスタ・ツー・即値フォーマットの等価物を有することは理解されよう。

移動命令（ＭＯＶ）は源レジスタ（ＲＳ）の内容が宛先レジスタ（ＲＤ）のなかへ移動されることを許す。

ロードレジスタ命令（ＬＲ）はＲＳにより指示されたメモリ位置の内容をＲＤへ移動する。もしＲＳがスタックポインタ（ＳＰ）であれば、ＳＰがロード以前にインクレメントされる。

記憶レジスタ命令（ＳＴＲ）はＲＳにより指示されたメモリ位置のなかへＲＤの内容を記憶する。もしＲＳがスタックポインタ（ＳＰ）であれば、ＳＰがデクレメントされる。

呼出し命令（ＣＡＬＬ）はＲＤ　により指示されたレジスタのなかへプログラムカウンタの内容＋２をロードする。ＲＳの内容は次いでプログラムカウンタのなかへ移動される。

移動及び集合状態命令（ＭＯＶＣ）はＲＳ及びＲＤの内容を移動する。

入力レジスタ命令（ＩＮＲ）はＩ／Ｏサイクルを規定する。ＲＳにより指示されたＩ／Ｏ位置の内容をＲＤへ移動する。他方、もしレジスタ・ツー・即値命令フォーマットがこの命令に対して規定されていれば、内部サイクルが指示され、それにより内部Ｉ／Ｏ位置がＲＤへ移動される。これらの内部位置の要約は下記の第４表に示されている。本明細書の現在の部分はＲＩＳＣの説明に向けられているけれども、ＲＩＳＣ、ＣＩＳＣ又は本発明の第二の局面からＲＩＳＣ又は本発明の第一の局面を記述的に完全に分離することは往々困難であり、実施例は双方を組み合わせることは理解されよう。第４表中に開示される機能の多くはＭＩＬ－ＳＴＤ－１７５０により指定されており、従ってまた本発明の第一の局面に厳密に関係付けられない。しかし、それらは結合性のために本明細書にこの点で開示されている。しかし、それらはその目的で理解されべきである。

第4表

コマンドフィールド (RS) (hex)	簡略記号	コマンド
10	-	予約されている
11	EA0	有効アドレス 0
12	EA1	有効アドレス 1
13	EA2	有効アドレス 2
14	PI	保留割込み
15	MK	マスクレジスタ
16	FT	故障レジスタ
17	SW	状態語
18	IRS	命令レジスタ 源フィールド
19	IRD	命令レジスタ
1A	-	予約されている
1B	-	予約されている
1C	PC	プログラムカウンタ
1D	PIPE	パイプレジスタ
1E	PCREL	PC相対的
1F	-	予約されている

１０．予約されている。

１１．有効アドレス　０（ＥＡ０）：ＲＩＳＣアーキテクチュアにより実現されないＭＩＬ－ＳＴＤ－１７５０アドレス指定モード、すなわち直接及び直接指標付け（Ｄ、ＤＸ）アドレス指定モードを実現するために使用される。ＭＩＬ－ＳＴＤ－１７５０命令レジスタ宛先フィールド（ＩＲＤ）により指示されるレジスタはパイプレジスタに加えられる。結果は選択されたアキュムレータ（Ａ）のなかに記憶される。ＭＩＬ－ＳＴＤ－１７５０パイプレジスタは次いでＭＩＬ－ＳＴＤ－１７５０プログラムカウンタにより指示されたメモリ位置の内容をロードされる。プログラムカウンタはポストインクレメントされる。

１２．有効アドレス　１（ＥＡ１）：ＲＩＳＣアーキテクチュアにより実現されないＭＩＬ－ＳＴＤ－１７５０アドレス指定モード、すなわち基底相対（Ｂ）アドレス指定モードを実現するために使用される。ＭＩＬ－ＳＴＤ－１７５０命令レジスタ基底フィールド（ＩＲＢ）により指示されるレジスタは命令レジスタ（ＩＲ）の下位８ビットに加えられる。結果はＡのなかに記憶される。

１３．有効アドレス　２（ＥＡ２）：ＲＩＳＣアーキテクチュアにより実現されないＭＩＬ－ＳＴＤ－１７５０アドレス指定モード、すなわち基底付き指標付き（ＢＸ）アドレス指定モードを実現するために使用される。ＭＩＬ－ＳＴＤ－１７

５０命令レジスタ基底フィールド（ＩＲＢ）により指示されるレジスタはＩＲＤフィールドにより指定されたレジスタに加えられる。結果はＡのなかに記憶される。

１４．保留割込み（ＰＩ）を読む：ＭＩＬ－ＳＴＤ－１７５０により指令されるが、ＲＩＳＣもしくはＣＩＳＣエミュレーションモードで使用される。保留割込みレジスタ（ＰＩ）の内容がＡのなかに記憶される。

１５．マスクレジスタ（ＭＫ）を読む：ＭＩＬ－ＳＴＤ－１７５０により指令されるが、ＲＩＳＣもしくはＣＩＳＣエミュレーションモードで使用される。マスクレジスタ（ＭＫ）の内容がＡのなかに記憶される。

１６．故障レジスタ（ＦＴ）を読む：ＭＩＬ－ＳＴＤ－１７５０により指令されるが、ＲＩＳＣもしくはＣＩＳＣエミュレーションモードで使用される。故障レジスタ（ＦＴ）の内容がＡのなかに記憶される。

１７．状態語（ＳＷ）を読む：ＭＩＬ－ＳＴＤ－１７５０モードに対して飲み使用される。ＭＩＬ－ＳＴＤ－１７５０状態語レジスタ（ＳＷ）の内容がＡのなかに記憶される。

１８．命令レジスタ源フィールド（ＩＲＳ）を読む：ＭＩＬ－ＳＴＤ－１７５０プログラムレジスタ２３０の４ビットＩＲＳフィールドがＡのなかに記憶される。

１９．ＩＲＤフィールド（ＩＲＤ）を読む：ＭＩＬ－ＳＴＤ－１７５０プログラムレジスタ２３０のなかの４ビットＩＲＤフィールドがＡのなかに記憶される。

１Ａ．予約されている。

１Ｂ．予約されている。

１Ｃ．プログラムカウンタ（ＰＣ）を読む：ＭＩＬ－ＳＴＤ－１７５０プログラムカウンタ２３４の内容がＡのなかに記憶される。

１Ｄ．ＭＩＬ－ＳＴＤ－１７５０命令パイプレジスタ（ＰＩＰＥ）を読む：ＭＩＬ－ＳＴＤ－１７５０命令パイプレジスタ２３２内容がＡのなかに記憶される。このパイプレジスタは次いでＭＩＬ－ＳＴＤ－１７５０プログラムカウンタにより指示されたメモリ位置の内容をロードされる。プログラムカウンタはポストインクレメントされる。

１Ｅ．プログラムカウンタ相対的（ＰＣＲＥＬ）を読む：ＭＩＬ－ＳＴＤ－１７５０命令レジスタ２３０のなかの８ビットフィールドを取り、それをＭＩＬ－ＳＴＤ－１７５０プログラムカウンタ２３４に加える。結果はＡのなかに記憶される。

１Ｆ．予約されている。

出力レジスタ命令（ＯＴＲ）はＲＳにより指示されたＩ／Ｏ位置のなかへＲＤの内容を記憶する。これはＩ／Ｏサイクルに相当する。しかし、もしレジスタ・ツー・即値命令フォーマットがこの命令に対して指示されるならば、内部サイクルが指示され、それにより内部Ｉ／Ｏ位置がＲＤへ移動される。これらの内部Ｉ／Ｏ位置の要約は下記の第５表に示されている。

第５表

コマンドフィールド（RS）(hex)	簡略記号	コマンド
10	CLFT	故障レジスタをクリア
11	PCL	プログラムカウンタロード
12	ENBL	割込みをイネーブル
13	DSBL	割込みをディスエーブル
14	SPI	保留割込みをセット
15	SMK	マスクレジスタをセット
16	SFT	故障レジスタをセット
17	SSW	状態語をセット
18	IRS	命令レジスタ源フィールド
19	IRD	命令レジスタ宛先フィールド
1A	INCS	IRSフィールドをインクレメント
1B	INCD	IRDフィールドをインクレメント
1C	IRL	命令レジスタをロード
1D	IRLD	命令レジスタをロード
1E	CCOFF	条件コードをディスエーブル
1F	RPI	保留割込みをリセット

１０．故障レジスタをクリア（ＣＬＦＴ）：ＲＩＳＣもしくはＭＩＬ－ＳＴＤ－１７５０エミュレーションモードで使用し得る。１６ビットの故障レジスタの内容が０にリセットされる。

１１．プログラムカウンタをロード（ＰＣＬ）：ＭＩＬ－ＳＴＤ－１７５０プログラムカウンタ２３４がＡの内容をロードされる。

１２．割込みをイネーブル（ＥＮＢＬ）：ＭＩＬ－ＳＴＤ－１７５０により指令されるが、ＲＩＳＣもしくはＣＩＳＣエミュレーションモードで使用される。このコマンドはマスクアウトされていない全ての割込みをイネーブルする。

１３．割込みをディスエーブル（ＤＳＢＬ）：ＭＩＬ－ＳＴＤ－１７５０により指令されるが、ＲＩＳＣもしくはＣＩＳＣエミュレーションモードで使用される。このコマンドはＤＳＢＬ命令の実行の開始時に（ディスエーブルされ得ないように定義されているものを例外として）全ての割込みをイネーブルする。一般にＩＮＴ０－電源喪失、ＩＮＴ１－機械エラー、及びＩＮＴ５－実行呼出しがディスエーブルされ得ない唯三つの割込みである。

１４．保留割込みレジスタをセット（ＳＰＩ）：ＭＩＬ－ＳＴＤ－１７５０により指令されるが、ＲＩＳＣもしくはＣＩＳＣエミュレーションモードで使用される。このコマンドはＡの１６ビットの内容を保留割込みレジスタへ出力する。もし割込みマスクの対応するビット位置に＂１＂が存在すし、（ＰＩ及びＭＫの双方のなかの同一のビット集合）、且つ割込みがイネーブルされれば、割込みは次の命令の実行の後に生起する。

１５．割込みマスクレジスタをセット（ＳＭＫ）：ＭＩＬ－ＳＴＤ－１７５０により指令されるが、ＲＩＳＣもしくはＣＩＳＣエミュレーションモードで使用される。このコマンドはＡの内容を割込みマスクレジスタへ転送する。対応するビット位置の＂１＂は割込みの生起を許し、また＂０＂は、マスクされ得ないものとして定義されているものを例外として、割込みの生起を阻止する。一般にＩＮＴ０－電源喪失、ＩＮＴ１－機械エラー、及びＩＮＴ５－実行呼出しがディスエーブルされ得ない唯三つの割込みである。

１６．故障レジスタをセット（ＳＦＴ）：ＭＩＬ－ＳＴＤ－１７５０により指令されるが、ＲＩＳＣもしくはＣＩＳＣエミュレーションモードで使用される。故障レジスタ（ＦＴ）の内容はＡの内容をロードされる。＂１＂のビット値は特定の故障をセットする。故障レジスタの必要条件に対するセクション４を参照。

１７．状態語をセット（ＳＳＷ）：ＭＩＬ－ＳＴＤ－１７５０状態語（ＳＷ）がＡの内容をロードされる。

１８．ＩＲＳ（命令レジスタ源フィールド）フィールドをロード（ＩＲＳ）：ＭＩＬ－ＳＴＤ－１７５０命令レジスタ２３０の４ビットのＩＲＳフィールドがＡのビット５～８をロードされる。

１９．ＩＲＤ（命令レジスタ宛先フィールド）フィールドをロード（ＩＲＤ）：ＭＩＬ－ＳＴＤ－１７５０命令レジスタ２３０の４ビットのＩＲＤフィールドがＡのビット１～４（ビット１はＬＳＢ）をロードされる。

１Ａ．ＩＲＳフィールドをインクレメント（ＩＮＣＳ）：ＭＩＬ－ＳＴＤ－１７５０のＰＲ２３０のなかの４ビットのＩＲＳフィールドがインクレメントされる。

１Ｂ．ＩＲＤフィールドをインクレメント（ＩＮＣＤ）：ＭＩＬ－ＳＴＤ－１７５０のＰＲ２３０のなかの４ビットのＩＲＤフィールドがインクレメントされる。

１Ｃ．命令レジスタをロード（ＩＲＬ）：ＭＩＬ－ＳＴＤ－１７５０のプログラムレジスタ２３０がパイプレジスタ２３２の内容をロードされる。ＲＩＳＣプログラムフローが（もし割込みが保留中であれば）割込みベクトルに、もしくは（ＭＩＬ－ＳＴＤ－１７５０のＰＲ２３０の上位８ビットにより規定される）マップベクトルに転送される。ＭＩＬ－ＳＴＤ－１７５０のパイプレジスタ２３０が次いでＭＩＬ－ＳＴＤ－１７５０のプログラムカウンタ２３４により指示されたメモリ位置の内容をロードされる。プログラムカウンタが２３４ポスト・インクレメントされる。条件付き状態フィールドがイネーブルされる。

１Ｄ．命令レジスタをロード（ＩＲＬＤ）：ＭＩＬ－ＳＴＤ－１７５０のプログラムレジスタ２

３０がパイプレジスタ２３２の内容をロードされる。ＲＩＳＣプログラムフローが（ＭＩＬ・ＳＴＤ・１７５０のＰＲ２３０の上位８ビットにより規定される）マップベクトルに転送される。ＭＩＬ・ＳＴＤ・１７５０のパイプレジスタ２３０が次いでＭＩＬ・ＳＴＤ・１７５０のプログラムカウンタ２３４により指示されたメモリ位置の内容をロードされる。プログラムカウンタ２３４がポスト・インクレメントされる。条件付き状態フィールドがイネーブルされる。

１Ｅ．条件コードをディスエーブル（ＣＣＯＦＦ）：ＭＩＬ・ＳＴＤ・１７５０の状態語（ＳＷ）の条件付き状態フィールドが、変更され得ないように、ディスエーブルされる。

１Ｆ．保留割込みをリセット（ＲＰＩ）：ＭＩＬ・ＳＴＤ・１７５０により指令されるが、ＲＩＳＣもしくはＣＩＳＣエミュレーションモードで使用される。Ａのなかのセットされた各ビットに対して、その対応する割込みビットがリセットされる。

飛越し・条件付きレジスタ命令（ＪＣＲ）は条件付き飛越し命令であり、もしＣＣフィールドに対応する論理＂１＂パターンがＣＣフィールドとＣＳフィールドとのビット・フォア・ビットのアンド演算の結果であるならば、即値フィールドがプログラムカウンタに加算される。５ビットの条件状態フィールドは参照符号＂ＶＣＰＺＮ＂を付されており、ここでＶはあふれ、Ｃは桁上げ、Ｐは正、Ｚは０、Ｎは負を意味する。それらはＡＬＵによりセット又はリセットされる。ＣＣフィールドはＲＩＳＣ命令レジスタ３４のＲＳフィールドである。種々のＣＣコマンドが第６表に示されている。

第６表

ＣＣフィールド	飛越し条件	簡略記号
NOP	•••	
00001	０より小さい	JC LT,RS
00010	０に等しい	JC EQ,RS
00011	０より小さい／０に等しい	JC LE,RS
00100	０より大きい	JC GT,RS
00101	０に等しくない	JC NE,RS
00110	０より大きい／０に等しい	JC GE,RS
00111	無条件	•••
10111	桁上げセット	JC CY,RS
01001	桁上げ又はＬＴ	JC CLT,RS
01010	桁上げ又はＥＱ	JC CEZ,RS
01011	桁上げ又はＬＥ	JC CLE,RS
01100	桁上げ又はＧＴ	JC CGT,RS
01101	桁上げ又はＮＥ	JC CNE,RS
01110	桁上げ又はＧＥ	JC CGE,RS
01111	無条件	•••
10000	あふれセット	JC V,RS
10001	あふれ又はＬＴ０	JC VLT,RS
10010	あふれ又はＥＱ０	JC VE,RS
10011	あふれ又はＬＥ０	JC VLE,RS
10100	あふれ又はＧＴ０	JC VGT,RS
10101	あふれ又はＮＥ０	JC VNE,RS
10110	あふれ又はＧＥ０	JC VGE,RS
10111	無条件	•••
11000	あふれ又は桁上げセット	JC VC, RS
11001	あふれ又は桁上げ又はＬＴ０	JC VCLT,RS
11010	あふれ又は桁上げ又はＥＱ０	JC VCEQ,RS
11011	あふれ又は桁上げ又はＬＥ０	JC VCLE,RS
11100	あふれ又は桁上げ又はＧＴ０	JC VCGT,RS
11101	あふれ又は桁上げ又はＮＥ０	JC VCNE,RS
11110	あふれ又は桁上げ又はＧＥ０	JC VCGE,RS
11111	無条件	•••

加算レジスタ命令（ＡＤＤ）はＲＳの内容をＲＤの内容に加える。結果はＲＤのなかに記憶される。あふれ条件は、もしオペランドが同一の符号を有し且つ結果が反対の符号を有するならば、生起する。

桁上げ付き加算レジスタ命令（ＡＤＤＣ）はＲＳの内容をＲＤの内容に加える。結果は、もし桁上げフラッグがセットされているならば、インクレメントされる。結果はＲＤのなかに記憶される。あふれ条件は、もしオペランドが同一の符号を有し且つ結果が反対の符号を有するならば、生起する。

加算バイト命令（ＡＢ）はＲＳの下位バイト内容をＲＤの下位バイト内容に加える。バイト結果はＲＤの下位バイトのなかに記憶される。あふれ条件は、もしバイトオペランドが同一の符号を有し且つバイト結果が反対の符号を有するならば、生起する。

加算レジスタ無符号命令（ＡＤＤＵ）はＲＳの内容をＲＤの内容に加える。結果はＲＤのなかに

記憶される。あふれ条件は影響されない。

減算レジスタ命令（ＳＵＢ）はＲＳの内容をＲＤの内容から差し引く。結果はＲＤのなかに記憶される。あふれ条件は、もしオペランドが反対の符号を有し且つ結果がＲＳと同一の符号を有するならば、生起する。

借り付き減算レジスタ命令（ＳＵＢＢ）はＲＳの内容をＲＤの内容から差し引く。結果は、もし桁上げフラッグがクリアされているならば、デクレメントされる。結果はＲＤのなかに記憶される。あふれ条件は、もしオペランドが反対の符号を有し且つ結果がＲＳと同一の符号を有するならば、生起する。

減算バイト命令（ＳＢ）はＲＳの下位バイト内容をＲＤの下位バイト内容から差し引く。バイト結果はＲＤの下位バイトのなかに記憶される。あふれ条件は、もしバイトオペランドが反対の符号を有し且つバイト結果がＲＳと同一の符号を有するならば、生起する。

比較レジスタ命令（ＣＭＰ）はＲＳの内容をＲＤの内容と比較する。もしＲＤがＲＳよりも大きいならば、Ｐ条件コードがセットされる。もしＲＤ＝ＲＳであれば、Ｚ条件コードがセットされる。もしＲＤがＲＳよりも小さいならば、Ｐ条件コードがセットされる。

論理アンドレジスタ命令（ＡＮＤ）はＲＳの内容とＲＤの内容とのアンド演算をする。結果はＲＤのなかに記憶される。

論理オアレジスタ命令（ＯＲ）はＲＳの内容とＲＤの内容とのオア演算をする。結果はＲＤのなかに記憶される。

論理排他的オアレジスタ命令（ＸＯＲ）はＲＳの内容とＲＤの内容との排他的オア演算をする。結果はＲＤのなかに記憶される。

論理否定レジスタ命令（ＮＯＴ）はＲＳの内容の１の補数をＲＤのなかに記憶する。

レジスタ内ビットリセット命令（ＲＢＲ）は、ＲＳのなかの値に従ってリセット（クリア）されるべきＲＤのなかのビットを選択する。下記の第７表には、ＲＳのなかの値とＲＤのなかでクリアされるビットとの関係が示されている。ＲＳのなかの最下位５ビットが、クリアされるべきビットを決定するのに使用される。他のビット値は重要でない。

第７表

ＲＳ内の値 MSB LSB	ＲＳ内でクリアされるビット	
0000 0000 0000 0000	31	MSB
0000 0000 0000 0001	30	
0000 0000 0000 0010	29	
⋮	⋮	
0000 0000 0001 1101	2	
0000 0000 0001 1110	1	
0000 0000 0001 1111	0	LSB

レジスタ内ビットセット命令（ＳＢＲ）は、ＲＳのなかの値に従ってセットされるべきＲＤのなかのビットを選択する。下記の第８表には、ＲＳのなかの値とＲＤのなかでセットされるビットとの関係が示されている。ＲＳのなかの最下位５ビットが、セットされるべきビットを決定するのに使用される。他のビット値は重要でない。

第８表

ＲＳ内の値 MSB LSB	ＲＤ内でセットされるビット	
0000 0000 0000 0000	31	MSB
0000 0000 0000 0001	30	
0000 0000 0000 0010	29	
⋮	⋮	
0000 0000 0001 1101	2	
0000 0000 0001 1110	1	
0000 0000 0001 1111	0	LSB

レジスタ内ビットテスト命令（ＴＢＲ）は、ＲＳのなかの値に従ってテストされるべきＲＤのなかのビットを選択する。下記の第９表には、ＲＳのなかの値とＲＤのなかでテストされるビットとの関係が示されている。ＲＳのなかの最下位５ビットが、テストされるべきビットを決定するのに使用される。他のビット値は重要でない。もしテストされるべきビットが０であれば、状態語のなかの２ビットがセットされる。

第9表

RS内の値 MSB LSB	RD内でテストされるビット	
0000 0000 0000 0000	31	MSB
0000 0000 0000 0001	30	
0000 0000 0000 0010	29	
≈	≈	
0000 0000 0001 1101	2	
0000 0000 0001 1110	1	
0000 0000 0001 1111	0	LSB

ロード命令レジスタ命令（ＬＲＩ）はＲＳにより指示された命令メモリ位置の内容をＲＤのなかに記憶する。

シフト論理レジスタ命令（ＳＬＲ）はＲＳの内容により選択されたビットの数だけＲＤの内容を論理的にシフトする。ＲＳの下位５ビットがシフトカウント及び方向を選択する。従って、シフト操作のために、可能なシフトの最大数は３２である。ＲＳの６ビットは通常又は拡張シフトを選択する。それがＲＳのビット１６と反対の符号である時、拡張シフトモードが選択される。拡張シフトモードでは、一時レジスタ（ＴＡ）はシフトされるべき値の上位３２ビットを含み、またレジスタ対ＲＤは下位３２ビットを含む。拡張シフトを使用して、６４ビットがシフトされ得るが、命令あたりは３２回のみである。下記の第１０表には、実行すべきシフトの方向及び数を決定する時にＲＳの値をどのように選択するかが示されている。

第10表

方向	通常 (RS) hex	シフトの廿	拡張 (RS) hex	シフトの廿
左	0000	1	0020	1
	0001	2	0021	2
	≈	≈	≈	≈
	001E	31	003E	31
	001F	32	003F	32
右	FFE0	32	FFC0	32
	FFE1	31	FFC1	31
	≈	≈	≈	≈
	FFFE	2	FFDE	2
	FFFF	1	FFDF	1

シフト算術レジスタ命令（ＳＡＲ）はＲＳの内容により選択されたビットの数だけＲＤの内容を算術的にシフトする。もしＲＤの符号が左シフトの間に変化すれば、あふれが生起する。ＲＳの下位５ビットはシフトカウント及び方向を選択する。従って、シフト操作のために、可能なシフトの最大数は３２である。ＲＳの６ビットは通常又は拡張シフトを選択する。それがＲＳのビット１６と反対の符号である時、拡張シフトモードが選択される。拡張シフトモードでは、一時レジスタ（ＴＡ）はシフトされるべき値の上位３２ビットを含み、またレジスタ対ＲＤは下位３２ビットを含む。拡張シフトを使用して、６４ビットがシフトされ得るが、命令あたりは３２回のみである。シフト論理レジスタ命令（ＳＬＲ）と結び付けて示された第１０表は、ＳＡＲ命令に対して、実行すべきシフトの方向及び数を決定する時にＲＳの値をどのように選択するかを示すのにも使用され得る。

シフトサイクルレジスタ命令（ＳＣＲ）はＲＳの内容により選択されたビットの数だけＲＤの内容を回転する。ＲＳの下位５ビットはシフトカウント及び方向を選択する。従って、シフト操作のために、可能なシフトの最大数は３２である。ＲＳの６ビットは通常又は拡張シフトを選択する。それがＲＳのビット１６と反対の符号である時、拡張シフトモードが選択される。拡張シフトモードでは、一時レジスタ（ＴＡ）はシフトされるべき値の上位３２ビットを含み、またレジスタ対ＲＤは下位３２ビットを含む。拡張シフトを使用して、６４ビットがシフトされ得るが、命令あたりは３２回のみである。ＳＬＲと結び付けて示された第１０表は、ＳＣＲ命令に対して、実行すべきシフトの方向及び数を決定する時にＲＳの値をどのように選択するかを示すのにも使用され得る。

乗算レジスタ命令（ＭＵＬＳ）はＲＳの内容によりＲＤの内容を乗算する。

移動レジスタ内バイト命令（ＭＯＶＢ）はＲＳの下位バイト内容をＲＤの下位バイト内容のなかへロードする。

スワップレジスタ内バイト命令（ＳＷＡＢ）はＲＳの下位及び上位バイト内容をそれぞれＲＤの上位及び下位バイト内容のなかへロードする。

除算レジスタ命令（ＤＩＶ）はＲＳの内容によりＲＤの内容を除算する。比はＲＤのなかに記憶される。残余はＴＡのなかに記憶される。

記憶レジスタ命令コード命令（ＳＴＲＩ）はＲＳにより指示された命令メモリ位置のなかへＲＤの内容をロードする。これは命令メモリサイクルである。

ＲＩＳＣ　ＣＩＳＣ

再び第３図を参照すると、状態語（ＳＷ）レジスタ、保留割込み（ＰＩ）レジスタ、割込みマスク（ＭＫ）及び故障レジスタ（ＦＴ）の形態でシステム状態を含んでいるＣＩＳＣに対する追加的な特徴がＩ／Ｏ１１０により得られることが思い出されよう。これらは全てＭＩＬ排他的オア回路ＳＴＤ排他的オア回路１７５０により定められている。ＲＩＳＣと結び付けて先に説明したように、それはオペランドデータをレジスタファイルに選択するのに使用されたことは思い出されよう。それは、ＣＩＳＣエミュレーションモードで、信号線２３８を経てレジスタファイルにＣＩＳＣプログラム命令を連係する追加的特徴を提供する。同様に、マルチプレクサ１２６は信号線２４０を経てレジスタファイルにＣＩＳＣプログラムアドレスを連係するのに使用される。

任意の所与のＣＩＳＣ応用プログラムに対して、プログラムカウンタ２３４は、類似の仕方での実行のためにパイプ２３２のなかに受信される各順次ＣＩＳＣ命令を通じてインクレメントする。各ＣＩＳＣ命令が受信されるにつれて、それは線２４２を経て制御装置３０ａに与えられ、それにより復号される。制御装置は、ＲＩＳＣ命令の群のなか第一のＲＩＳＣ命令が線２４２を経て受信された特定のＣＩＳＣ命令をエミュレートするため第２図のＰＲＯＭ２０のなかに記憶されている対応するＲＩＳＣアドレスを"ルックアップ"する。このＲＩＳＣの第一のアドレスが信号線２４４上に置かれ、またＲＩＳＣ命令カウンタ３２に連係するのにも使用される。エミュレーションのために使用されるＲＩＳＣに対して、ＣＩＳＣプログラムレジスタ２３０が、現在実行されているＣＩＳＣプログラム命令を保持するのに使用される一時レジスタとして設けられている。これらの命令はＣＩＳＣ命令"パイプ"２３２から受信される。このパイプ２３２は、ＣＩＳＣプログラムカウンタによりアドレス指定されてバッファ１０６、信号線１０８及びＩ／Ｏ１１０を経て与えられるＣＩＳＣ命令をデータバス１８を経て受信する。ＣＩＳＣプログラム命令アドレスは、対応するＣＩＳＣプログラム命令をパイプ２３２のなかへロードする以前にＣＩＳＣプログラムレジスタ２３４のなかに一時記憶される。プログラムカウンタはそのＣＩＳＣ命令アドレスを、制御装置３０ａにより制御されるバッファ２３６を経て命令バス１６上に与える。

基本ＥＩＳＣのマルチプレクサ１２４が、算術及び論理演算のため及びレジスタファイルへの命令データの連係のためにＡＬＵへの第一の入力を与えられ、そこから次いでＲＩＳＣ命令アドレスバス１２上に置かれる。所望のＲＩＳＣ命令が次いで命令バス１４上に現れ、また前記のように復号且つ実行される。命令カウンタは次いでこの機械サイクルの中央で、すなわち制御装置３０ａにより制御されてエミュレーション群の開始時にＲＩＳＣ命令をアドレス指定するためＣＫ３の立ち上がり縁でインクレメントされる。クロック線２４６は、ＣＫ３の立ち上がり縁で高レベルになる信号線を示す。命令カウンタはエミュレーション群の第一の命令以外の全ての命令に対してＣＫ１でインクレメントされる。

群のなかの第一のＲＩＳＣエミュレーション命令で命令カウンタをロードする以前の機械サイクルの間、パイプ２３２及びプログラムレジスタ２３０は、それらのそれぞれの内容を適当な宛先に、すなわちプログラムレジスタ２３０の内容を制御装置３０ａに、またパイプ２３２の内容を（もし適当であれば）マルチプレクサ１２４に与える目的でＣＫ１の立ち上がり縁でクロックされてい

る。すなわちこのＣＫ１の立ち上がり縁でプログラムカウンタ２３４がクロックされる。こうして、特定の機械サイクルの間にＣＩＳＣ２２のなかへクロックされる新しいＣＩＳＣ命令に対して、対応する最初のＲＩＳＣ命令アドレスがＲＩＳＣ命令カウンク３２のなかへロードされる以前に半サイクルの遅延が存在する。

基本ＲＩＳＣと結び付けて以上に説明された命令集合を使用してこの仕方で実行されるＣＩＳＣの例は後で示される。追加的な例は不必要である。なぜならば、熟練したプログラマーは、ＣＩＳＣ命令の集合を与えられて、ＲＩＳＣエミュレーション群の対応する集合を容易に構成し得るからである。

例として選択されるＣＩＳＣ命令集合は、命令コードの一つが直接アドレス指定モードのなかの単精度整数加算であるＭＩＬ－ＳＴＤ－１７５０Ａ命令集合である（１９８０年７月２日付けのＭＩＬ－ＳＴＤ－１７５０Ａの第８９頁の5.5.5節を参照）。そのモードのなかで、加算のためには、導き出されたオペランドがＲＡレジスタの内容に加算される。結果（２の補数和）はレジスタＲＡのなかに記憶される。フォーマットは下に示されている。

アドレスモード	簡略記号	フォーマット／命令コード			
		8	4	4	16
D	A RA,ADDR	AO	RA	RX	ADDR

８ビットの命令コードは"加算直接"演算コードである。４ビットのＲＡフィールドは、導き出されたオペランドが加算され且つ結果が記憶されるレジスタを指定する。ＲＡフィールド（４ビット）はインデックスレジスクである。１６ビットの加算器フィールドは導き出されたオペランドを得るために設けられている。こうして、示されている命令は、第２図の命令バス２４がアクセスしたＭＩＬ－ＳＴＤ－1750によるメモリ空間のなかに記憶され得る。それはデータバス１８を経て与えられ、また上位１６ビットは復号のために線２４２を経て制御装置３０ａに与えるためプログラムレジスタ２３０のなかへ与えられ、また下位１６ビット（ＡＤＤＲ）はパイプ２３２に与えられる。もちろん、示されている実施例では、これらの命令が受信されるオペラントデータバスは単に１６ビットのバスであり、上半及び下半が別々に送られなければならないことは理解されよう。いずれの場合にも、プログラムレジスタ２３０のなかの上半は線２４２を経て制御装置３０ａに与えられ、そこで復号され、また特定のＣＩＳＣ命令をエミュレートするためＲＩＳＣエミュレーション群の開始アドレスを示す適当なＲＩＳＣアドレスが線２４４を経てＲＩＳＣ命令カウンク３２に与えられる。

ＭＩＬ－ＳＴＤ－１７５０Ａの単精度整数加算をエミュレートするための第一のこのようなＲＩＳＣ命令は下記の通りである：

ＩＮＲ　Ａ０，ＥＡ１

上記の命令に従って、パイプ２３２のなかに留まっている下位１６ビットのアドレスが次いでマルチプレクサ１２４及びＡＬＵ１０４を経てレジスタファイル７４のレジスタＡ０のなかへ送られる。

命令カウンク３２が次いで下記のような次のエミュレーション命令をアドレス指定するためインクレメントされる：

ＬＲ　Ａ１，Ａ０

上記のロードレジスク命令は、Ａ０のなかに留まっているアドレスにより指示されるメモリ位置の内容がＡ１へ移動されるメモリサイクルである。この群のなかで実行されるべき次のＲＩＳＣエミュレーション命令は下記の通りである：

ＡＤＤ　ＩＲＳ，Ａ１

上記の加算レジスタ命令はＰＲ"ｓ"フィールド（４ビット）により指示されたレジスタの内容を取り、またそれをレジスクＡ１の内容に加える。加算の結果はＰＲ"ｓ"フィールド（１６の"Ｒ"レジスタの一つ）により指示されたレジスタのなかに記憶される。この群に対する次の最後のＲＩＳＣエミュレーション命令は、基本的に言って、パイプ２３２及びレジスタ２３０からの次のＣＩＳＣ命令を取る。それは下記の通りである：

OTR A0, IRL

こうして、MIL-STD-1750Aの命令の各々が第2図のPROM20のなかに記憶されているRISC命令の群としてエミュレートされ得る。プログラミングの分野の当業者にとって、それぞれ複合命令集合の命令の一つをエミュレートするためのRISC命令の種々の群を構成することは簡単である。従って、RISCコードでエミュレートされる全てのMIL-STD-1750Aの命令の例はここには示されない。示された例で十分である。

MIL-STD-1750CISCをエミュレートするためのRISCエミュレータを設計するために本アプローチを使用することにより特別な追加的な利点が得られる。第2図に示されている2チップ解決では、PROM20がユーザー固有の特別なユーザーにより定義された機能を定義するのにユーザーにより使用され得る。1986年9月29日付けのMIL-STD-1750Bのドラフトバージョンの第147頁には、"特殊"アドレス指定モードに対するビルトイン機能命令が記載されている。この命令はユーザーにより定義された特殊な演算を呼び出す。命令はそれにすぐ続く一つ又はそれ以上の追加語を使用し得る。その数及び解釈は8ビットの命令コード拡張により決定される。市販品のMIL-STD-1750チップに対してこのようなビルトイン機能を与えるための現在のアプローチは、特別に要求されるビルトイン機能を組み入れるため顧客メイドのアプローチに従ってシリコン上に縮小されなければならない製造者からの特製チップを顧客が特別に注文することである。これは非常に費用がかかるアプローチであり、費用が制約されている多くの応用にはオプションであり得ない。しかし、本発明のアプローチを使用すれば、ユーザーはRISC命令集合の可能性を知らされ、またユーザー固有のビルトイン機能をPROM20に格納するためのソフトウェアのなかにプログラムし得る。こうしてユーザーは、RISCハードウェア10のなかで実行されるRISCソフトウェアを使用してユーザー固有のビルトイン機能を非常に安価に製造し得る。これは従来のアプローチを使用しては可能でなかった非常に望ましい特徴である。

設計方法

本発明の第三の局面によれば、複合命令集合計算機（CISC）アーキテクチュアを実現するための設計方法が提供される。CISCを設計する現在の方法はいくつかの異なる形態をとる。第一のアプローチでは、単一レベル制御を使用して全ての命令が実現される。換言すれば、実行は例えばZ8000に使用されているようなハードウェアにより制御される。第9図には、CISC命令集合アーキテクチュア300が"ハード配線による"制御装置302を介して実現されるこのようなアプローチが示されている。

他のアプローチでは、2レベル制御を使用して全ての命令が実現される。換言すれば、実行はハードウェアのマイクロコード（ファームウェア）制御により制御される。第10図には、信号線306上の命令を介してファームウェア308へ供給されるCISC命令集合アーキテクチュア304が示されており、ファームウェア308が残余の実行機能を信号線312を経てハードウェア制御装置310へ通す。このようなアプローチの例はモトローラ68000である。

別のアプローチでは、たいていの命令は2レベル制御を使用して実行され、残余の命令はソフトウェアを使用して実行される。このアプローチは第11図に示されており、CISC命令集合アーキテクチュア31.4に属する複合命令はファームウェア制御316とソフトウェアエミュレーション318を供給するための手段との双方に供給される。ファームウェアは信号線320上の複合命令とエミュレーションユニット318で行われたソフトウェアエミュレーションの結果を示す線322上の信号とに応答する。残余の制御機能はハード配線による制御装置324のなかで実行される。このような実現の例はマイクロヴァックスである。

第12図には、複合命令集合計算機アーキテク

チュアを実現するための本発明による方法が示されている。このアプローチでは全ての命令が2レベル制御を使用して実行され、実行はハードウェアのソフトウェア制御により制御される。例はＲＩＳＣによりエミュレートされるＭＩＬ－ＳＴＤ－１７５０ＡのＣＩＳＣに対して先に示されている。ソフトウェアエミュレーションは、従来の場合のように、ソフトウェアインタプリタを必要とする。ソフトウェア解釈は遅い。さらに、全ての命令は実行され得ない。ここに開示される方法はＭＩＬ－ＳＴＤ－１７５０、ＶＡＸ、ＮＥＢＵＬＡなどを含む多数のＣＩＳＣ命令集合に適用される。このアプローチでは、先ずＲＩＳＣ設計フィロソフィを使用して（ハード配線による）単一レベル制御が構成される。そうする間に、設計者はＲＩＳＣ（ハード配線）命令集合の実行を最大化するべく試みる。いったんＲＩＳＣがハードウェア設計されると、それはシリコンへの縮小のために工場に送られ得る。設計者は次いで、上記の例で説明したように、ＲＩＳＣ命令を使用してＣＩＳＣエミュレータを書く。このアプローチの合理性はＲＩＳＣ設計時間がＣＩＳＣ設計時間よりもはるかにはるかに少ないことである。例えば、フェアチャイルドＦ９４５０、ＭＤ２８１は開発に3年以上も要したことが知られている。本アプローチを使用すれば、ＭＩＬ－ＳＴＤ－１７５０のＲＩＳＣエミュレータは1年以内で開発され、また認可を得るのにシリコン工場への送付を1回しか必要としなかった。

第１２図には本発明のアプローチを使用する設計方法が示されている。ＣＩＳＣ命令集合アーキテクチュア３２６は信号線３２８を経て、ハード配線による装置３３４のなかでのハードウェア実行のために線３２３上に信号を供給するソフトウェアエミュレーションユニットへ複合命令を供給する。このアプローチの主要な利点は、（１）ソフトウェアの速度に対するハードウェアの複雑さのかねあい、及び（２）ハードウェアの設計時間（通常はソフトウェアの設計時間よりもはるかに長い時間を要する）の短縮である。こうして、最初の設計からワーキングコントロールまでの時間がはるかに少なくてすむ。

第１３図は本発明の第三の局面による設計ステップを示すブロック図である。ＲＩＳＣは先ずステップ３４０で設計される。いったんハードウェア設計及びＲＩＳＣ命令集合が決められると、次のステップ３４２は製造のための設計に送られる。同時に、第2図のＰＲＯＭ２０上で一緒に群にされているＲＩＳＣ命令で選択されたＣＩＳＣ命令集合をエミュレートするためエミュレーションコードが書かれるステップ３４４が同時に実行される。

以上に於ては本発明を特定の好ましい実施例について説明してきたが、本発明はこれらの実施例に限定されるものではなく、本発明の範囲内にて種々の実施例が可能であることは当業者にとって明らかであろう。

4．図面の簡単な説明

第1図は本発明の第一の局面によるＲＩＳＣ１０のブロック図である。

第2図は本発明の第二の局面によりデータバス１８を経て複合命令の源から受信された複合命令をエミュレートするためのＲＩＳＣ命令の群を含んでいる別個のメモリストア２０とならんで使用される本発明の第一の局面によるＲＩＳＣ１０のブロック図である。

第3図は本発明の第一の局面によるＲＩＳＣ１０を一層詳細に細線で示し、また本発明の第二の局面によりＣＩＳＣをエミュレートし得るＲＩＳＣを実行するために必要な追加的ハードウェアを太線で示すブロック図である。

第4図は第3図中に示されている信号線上の種々の信号の電圧レベルにを示す種々の波形図である。

第5図は第3図のレジスタファイル７４を一層詳細に示すブロック図である。

第6図は第3図の源モジュール７８を一層詳細に示すブロック図である。

第7図は第3図の宛先モジュール７６を一層詳細に示すブロック図である。

第８図は第３図のアキュムレータモジュール１４０を一層詳細に示すブロック図である。

第９図は全ての命令が単一レベル制御を使用して実行されるＣＩＳＣの従来の設計実行を示すブロック図である。

第１０図は全ての命令が二レベル制御を使用して実行されるＣＩＳＣの設計への従来のアプローチを示すブロック図である。

第１１図は大部分の命令が二レベル制御を使用して実行され、また残りの命令がソフトウェアを使用して実行されるＣＩＳＣの設計への従来のアプローチを示すブロック図である。

第１２図は本発明の第三の局面により全ての命令が二レベル制御を使用する設計アプローチを使用するＣＩＳＣ実行を示すブロック図である。

第１３図は本発明の第三の局面によりＣＩＳＣをエミュレートするためのＲＩＳＣを設計・製造する際に遂行され得るステップを示すブロック図である。

１０…信号プロセッサ（ＲＩＳＣ）、１２…ＲＩＳＣ命令アドレスバス、１４…縮小命令バス、１６…オペランドアドレスバス、１８…データバス、２０…メモリ（ＰＲＯＭ）、３０…制御装置、３２…命令カウンタ、３４…命令レジスタ、７４…レジスタファイル、７６…宛先モジュール、７８…源モジュール、１０４…ＡＬＵ、１２４、１２６…マルチプレクサ、１４０…アキュムレータモジュール、１５０、１５２、１５８、１６０…マルチプレクサ、２００…シフタマルチプレクサ、２０４…宛先レジスタ、２１０…アキュムレータモジュール、２１２…マルチプレクサ、２２２…シフタマルチプレクサ、３００…ＣＩＳＣ命令集合アーキテクチュア、３０２…ハード配線による制御装置、３０４…ＣＩＳＣ命令集合アーキテクチュア、３０８…ファームウェア制御ユニット、３１０…ハードウェア制御装置、３１４…ＣＩＳＣ命令集合アーキテクチュア、３１６…ファームウェア制御ユニット、３１８…エミュレーション装置、３２４…ハードウェア制御装置、３２６…ＣＩＳＣ命令集合アーキテクチュア、３３０…ソフトウェアエミュレーションユニット、３３４…ハード配線による制御装置

特許出願人　ユナイテッド・テクノロジーズ・コーポレイション

代　理　人　弁理士　　明　石　昌　毅

図面の簡単な

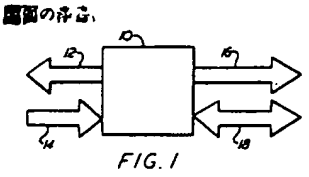


FIG. 1

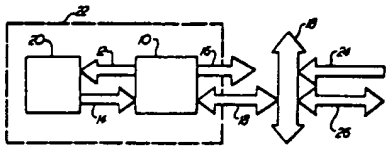


FIG. 2

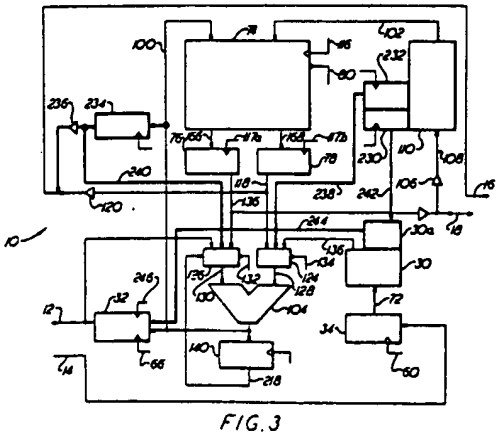


FIG. 3

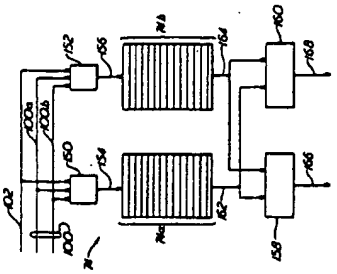


FIG. 5

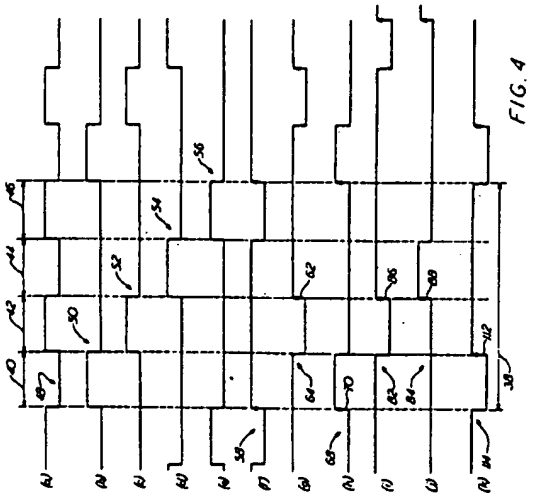


FIG. 4

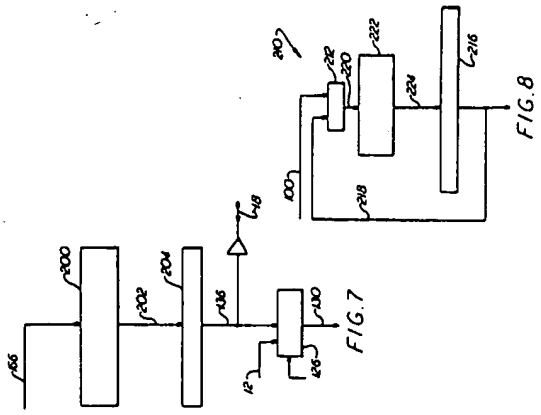


FIG. 8

FIG. 7

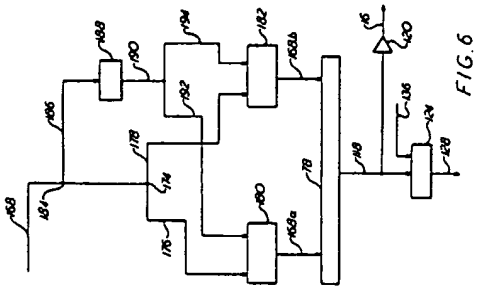


FIG. 6

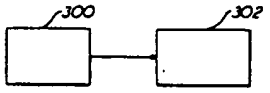


FIG. 9

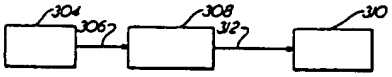


FIG. 10

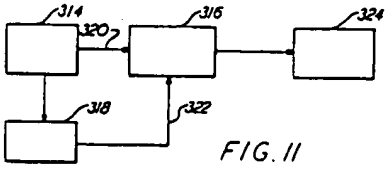


FIG. 11

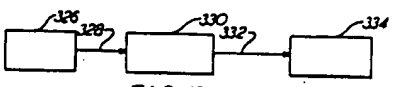


FIG. 12

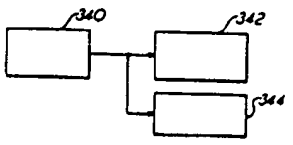


FIG. 13

(方 式) (自 発)

手 続 補 正 書

昭和63年1月27日

特許庁長官　殿

1. 事件の表示　昭和62年特許願第317311号

2. 発明の名称　信号処理方法、信号プロセッサ、その設計方法及びマイクロプロセッサ

3. 補正をする者

事件との関係　特許出願人

住 所　アメリカ合衆国コネチカット州、ハートフォード、フィナンシャル・プラザ 1

名 称　ユナイテッド・テクノロジーズ・コーポレイション

4. 代 理 人

居 所　〒104 東京都中央区新川1丁目5番19号 茅場町長岡ビル3階 電話551-4171

氏 名　(7121) 弁理士 明 石 昌 毅

5. 補正の対象　図面、優先権証明書及び訳文

6. 補正の内容　別紙の通り

方 式

特許庁 63. 1. 28